NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A PERSON-TRACKING MOBILE ROBOT USING AN ULTRASONIC POSITIONING SYSTEM

by

Chuan-Hao Yang

December 2005

Thesis Advisor:	Xiaoping Yun
Second Reader:	Marcello Romano

Approved for public release, distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: A Person-tracking Mobile Robot Using An Ultrasonic Positioning System			**5. FUNDING NUMBERS**
6. AUTHOR(S) Chuan-Hao Yang			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release, distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

It is desirable in many applications that a mobile robot is able to track and follow a person. There have been various efforts in literature to create person-tracking robots. However, current person-tracking robots are not capable of operating in unstructured environments.

The problem of creating a person-tracking mobile robot has been studied by many researchers in literature. There are two main issues associated with this problem. The first issue is to equip a robot with proper sensory devices so that it is able to identify and locate the target person in a crowd in real time. Various approaches have been investigated, including vision, infrared sensors, ultrasonic sensors, and other approaches. The second issue is to control and navigate the robot so that it follows the target person within a certain distance. This seems to be simple, but in reality it is a fairly difficult task. For example, if the target person is in a busy corridor with many people standing and walking, the robot has to constantly avoid other people while following the target. There is still no reported evidence that a person-tracking robot has been implemented that is able to track a person in arbitrary environmental conditions.

In this research, by using an innovative RF/ultrasonic sensor system, an intelligent person-tracking mobile robot is to be implemented that is able to follow the target person in unstructured, practical environments. The main focus of the thesis is development and implementation of control algorithms.

14. SUBJECT TERMS Person-Tracking, Person-Following, Obstacle Avoidance, Potential Field, Intelligent Space, Combined Force, Resultant Force, Attractive Force, Motion Planning, Unstructured Environment			**15. NUMBER OF PAGES** 71
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release, distribution is unlimited

A PERSON-TRACKING MOBILE ROBOT USING AN ULTRASONIC POSITIONING SYSTEM

Chuan-Hao Yang
Captain, Taiwan Army
B.S., National Defense University, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2005

Author: Chuan-Hao Yang

Approved by: Xiaoping Yun
Thesis Advisor

Marcello Romano
Second Reader

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

It is desirable in many applications for a mobile robot to track and follow a person. There have been various efforts in literature to create person-tracking robots. However, current person-tracking robots are not capable of operating in unstructured environments.

The problem of creating a person-tracking mobile robot has been studied by many researchers in literature. There are two main issues associated with this problem. The first issue is to equip a robot with proper sensory devices so that it is able to identify and locate the target person in a crowd in real time. Various approaches have been investigated, including vision, infrared sensors, ultrasonic sensors, and other approaches. The second issue is to control and navigate the robot so that it follows the target person within a certain distance. This seems simple, but in reality it is a fairly difficult task. For example, if the target person is in a busy corridor with many people standing and walking, the robot has to constantly avoid other people while following the target. There is still no reported evidence that a person-tracking robot has been implemented that is able to track a person in arbitrary environmental conditions.

In this research, by using an innovative RF/ultrasonic sensor system, an intelligent person-tracking mobile robot is to be implemented that is able to follow the target person in unstructured, practical environments. The main focus of the thesis is development and implementation of control algorithms.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
- **A. PERSON-TRACKING MOBILE ROBOT....1**
- **B. MOTIVATION....2**
- **C. SEVERAL APPROACHES TO THE PERSON-TRACKING ROBOT....2**
  - **1. Vision-Based Approach....2**
  - **2. Non-vision Based Approach....3**
  - **3. Transmitter-and-receiver Based Approach....3**
    - ***a. Person Tracking Using Blinking LED Devices....3***
    - ***b. Person Tracking Using an Ultrasonic Positioning System....4***
  - **4. Intelligent Space Approach....4**
  - **5. Combined/Multi-Modal Approach....4**
- **D. THESIS OBJECTIVES....6**
- **E. THESIS OUTLINE....6**

II. SYSTEM ARCHITECTURE....7
- **A. ULTRASONIC POSITIONING SYSTEM....7**
- **B. ROBOT SYSTEM....10**
  - **1. Bumper Sensors....11**
  - **2. Sonar Sensors....11**
  - **3. Motor/Motion Sensors....13**
- **C. INTERACTION BETWEEN ULTRASONIC POSITIONING SYSTEM AND SONAR SENSORS....13**
- **D. SUMMARY....15**

III. ALGORITHM....17
- **A. POTENTIAL FIELD ALGORITHM....17**
  - **1. Robot Coordinate....17**
  - **2. Attractive Forces Derived From the Readings of the Ultrasonic Sensor....17**
  - **3. Attractive Forces Derived From the Readings of Sonar Sensors....18**
  - **4. Potential Field Motion Planning From Combined Forces....20**
  - **5. Resultant Forces to Translation and Steering Velocities Conversion....21**
- **B. ALGORITHM USED AS THE TARGET IS IN A CERTAIN RANGE....22**
- **C. OBSTACLE AVOIDANCE ALGORITHM....23**
  - **1. Motion Planning for Obstacles on the Right Forward of the Robot....24**
  - **2. Motion Planning for Obstacles on the Left Forward of the Robot....25**
  - **3. Motion Planning for Obstacles in Front of the Robot....26**
- **D. OVERALL ALGORITHM OF A PERSON-TRACKING MOBILE ROBOT....28**

E. SUMMARY 29

IV. EXPERIMENTS AND RESULTS 31
A. PERSON-TRACKING IMPLEMENTATION WITHOUT OBSTACLES....................31
B. PERSON-TRACKING IMPLEMENTATION WITH AN OBSTACLE BETWEEN THE ROBOT AND THE TARGET PERSON 32
C. PERSON-TRACKING IMPLEMENTATION WHEN THE TARGET PERSON MAKES A TURN AT A CORNER....................33
D. PERSON-TRACKING IMPLEMENTATION IN AN UNSTRUCTURED ENVIRONMENT 35
E. SUMMARY 36

V. CONCLUSION AND FUTURE WORK 39
A. CONCLUSION 39
B. FUTURE WORK....................40

APPENDIX....................41

LIST OF REFERENCES....................49

INITIAL DISTRIBUTION LIST....................53

LIST OF FIGURES

Figure 1. Person-Tracking Mobile Robot using an Ultrasonic Positioning System. 5
Figure 2. Specially Made Vest and Ultrasonic Transmitter. 5
Figure 3. Ultrasonic Positioning System. 8
Figure 4. The Arrangement of the Bumper Sensors. 11
Figure 5. The Arrangement of the Sonar Sensors. 12
Figure 6. The Processing of a Sonar Sensor. 12
Figure 7. The Values of the Motor State Vector Bitmap. 13
Figure 8. Process Timing Diagrams of Sonar and Ultrasonic Devices. 14
Figure 9. Robot Coordinate. 17
Figure 10. Attractive Forces Derived From the Readings of the Ultrasonic Sensor. 18
Figure 11. Distances Measured by Sonar Sensors. 19
Figure 12. Combined Force Derived From the Readings of the Sonar Sensors. 19
Figure 13. Resultant of the Attractive Forces. 20
Figure 14. Motion Planning for Obstacles on the Right Forward of the Robot. 24
Figure 15. Motion Planning for Obstacles on the Left Forward of the Robot. 25
Figure 16. Motion Planning for Obstacles in Front of the Robot $(\gamma < 0)$ 26
Figure 17. Motion Planning for Obstacles in Front of the Robot $(\gamma \geq 0)$. 27
Figure 18. Overall Algorithm Flowchart. 28
Figure 19. Robot Trajectory in an obstacle-free situation. 31
Figure 20. Robot Trajectory when encountering an obstacle. 32
Figure 21. Robot Trajectory when there is an obstacle between the robot and the target person in the beginning of the person-tracking task. 33
Figure 22. Robot Trajectory when the target person makes a turn at a corner. 34
Figure 23. Robot Trajectory in an unstructured environment. 36

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. The State Vectors of the Robot System. 10
Table 2. Timing Parameters. 14

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to acknowledge the constant guidance and instruction of my advisor, Professor Xiaoping Yun, and second reader, Professor Marcello Romano, the technical assistance and support from our laboratory engineer, James Calusdian, and President of SJAutomation L.L.C., Steven Jacobs, and the moral support from my friends and overseas family.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

A person-tracking mobile robot is an innovative mobile robot, which is able to perform person-following and obstacle avoidance tasks simultaneously. Current person-tracking robots are not fully capable of operating in unstructured environments. The objective of this thesis is to develop a person-following mobile robot capable of operating in unstructured or semi-structured environments.

The problem of creating a person-tracking mobile robot has been studied by many researchers in literature. There are two main issues associated with this problem. The first issue is to equip a robot with proper sensory devices so that it is able to identify and locate the target person in a crowd in real time. Various approaches have been investigated, including vision, infrared sensors, ultrasonic sensors, and other approaches. The second issue is to control and navigate the robot so that it follows the target person within a certain distance. This seems simple, but in reality it is a fairly difficult task. For example, if the target person is in a busy corridor with many people standing and walking, the robot has to constantly avoid other people while following the target person. There is still no reported evidence that a person-tracking robot has been implemented that is able to track a person in arbitrary environmental conditions.

An RF/ultrasonic positioning system is utilized in this thesis for identifying and locating the target person. This system includes two ultrasonic receivers along with an RF transmitter installed on the top of the robot, and eight ultrasonic transmitters along with an RF receiver carried by the target person. The function of this system is to measure the relative position of the target person in terms of ranges and bearings in real time. The target information serves as a part of the control inputs to the robot system in performing the person-tracking task.

The mobile robot has a sonar system that includes 16 sonar sensors arranged in a ring. The function of this system is to measure the distance and the direction of obstacles. The information of the obstacle from the sonar sensors is another part of the inputs. By

using this information, the robot is able to avoid the obstacles encountered during the person-tracking task.

The overall algorithm used in this thesis includes two major sub-algorithms, the potential field algorithm and the obstacle avoidance algorithm. By regarding the readings from the RF/ultrasonic positioning system and the sonar system as the attractive forces, the potential field algorithm is to compute the resultant force from those attractive forces, and furthermore convert it into translation velocity and steering velocity, which control the motion of the robot. The obstacle avoidance algorithm is executed when the robot is too close to the obstacles.

Four main experiments are conducted to validate the person-tracking ability of the mobile robot using an RF/ultrasonic positioning system. The first experiment is conducted to verify the normal function of the mobile robot using a direct person-tracking condition without any obstacle between the target person and the robot. The second experiment is designed to add the obstacle in the procedure of person-tracking task, and examine the ability of the robot to implement obstacle avoidance and person-tracking simultaneously. The third experiment is for examining the ability of the robot in a situation where the robot needs to maintain tracking of the target person during a turn at a corner. The fourth experiment is based on the examination of the robot behavior in a general environment, which is unstructured.

Based on experiment results, the feasibility of developing a person-tracking mobile robot system using an RF/ultrasonic positioning system is established.

I. INTRODUCTION

A. PERSON-TRACKING MOBILE ROBOT

A person-tracking mobile robot is a robot that follows a specific person while simultaneously implementing obstacle avoidance. The robot follows only the target person and regards all other objects as obstacles. This means that even if there are several people walking around the environment, the robot should follow this specific person and avoid others. Therefore, any instances of the robot following the wrong person should be handled in the implementation of person-tracking.

A Nomad 200 mobile robot with an additional RF/ultrasonic positioning system has been adopted to implement person-tracking. The first stage is person positioning. Ultrasonic signals are transmitted from several transmitters located on a specially made vest, which the target person wears. The signals are received by two receivers located on the top of the robot. Those signals are then processed to produce a part of the control inputs, which are in the form of distances and angles, to the robot algorithm. In the second stage, the robot is equipped with 16 sonar rangefinders in a ring, which has a 22.5 degree angle between every two adjacent units. Those 16 sonar units transmit the sonar waves and receive the echoes sequentially to compute the distance between robot and obstacles in every direction. Finally, using the 16 distance data along with the readings from the ultrasonic positioning system, a potential field based motion algorithm [1] is formed. (The potential field based motion algorithm will be discussed in Chapter III.) Additionally, several specific sub-algorithms will be implemented when the robot is too close to the obstacles. Therefore, person-tracking and obstacle avoidance can both be implemented concurrently in unstructured environments.

The 16 sonar rangefinders and RF/ultrasonic positioning system operate using similar acoustic principles with partial differences. In order to avoid confusion, the original 16 rangefinders are named “sonar rangefinders,” “sonar sensors,” or “sonar units,” and the additional positioning system is named “ultrasonic positioning system,” “ultrasonic sensor,” or “ultrasonic unit” in this thesis. More detailed description will be illustrated in Chapter II.

B. MOTIVATION

It is desired in many applications that the mobile robot be able to track and follow a person. There have been various efforts in literature to create person-tracking robots. However, current person-tracking mobile robots are not capable of operating in unstructured environments. Because several main approaches, such as vision and infrared sensors, are not fully reliable in all situations, it is necessary to explore other methods. The main objective of this research is to investigate the feasibility of developing a person-tracking robot system using an RF/ultrasonic positioning system.

C. SEVERAL APPROACHES TO THE PERSON-TRACKING ROBOT

1. Vision-Based Approach

This is an approach using a camera to capture the image of the target person. The image has to be updated in real-time. This method assumes that the target person detection is successful, although this may always be a challenge. After detecting the target person in an image, the control information, including directions and distances, will be computed from the variations of the target position and size in the image. The robot should then be able to move toward the target person based on this information. Numerous researches [2-15] have adopted and adapted this approach to develop the person-following mobile robots. However, several uncertainties can still be significant enough to influence the efficiency of target detection. One factor that affects the detection is light condition. Determining the target person in the image can be relatively more difficult when the color or brightness of the target is not outstanding enough to make it different from that of the background or other obstacles. Another factor that affects detection is the simultaneous motion of both person and robot. The vision sensor can easily lose the target person when the target person moves too fast. Some researchers used active cameras. This reduced the problem of losing the target person, but increased the difficulty in the algorithm design. This approach is not suitable for the robot to perform obstacle avoidance. It is difficult for the robot to tell the difference between the target person and other obstacles. The situation will only be worse when there are several persons moving around in the same environment. It is possible and likely for the robot to lose the target person if the environment is unstructured.

2. Non-vision Based Approach

A non-vision based approach uses several kinds of rangefinders, such as sonar sensors, infrared sensors, and others. Each rangefinder on the robot can determine the distance between the nearest object and the rangefinder itself. Because the robot is not able to distinguish between object and target person, this approach can only be adopted to implement either obstacle avoidance when regarding all the objects as obstacles or person-tracking when the target person is always the nearest object to the robot without any obstacle in between. Using a Nomad 200 mobile robot equipped with 16 sonar rangefinders, the distance of the object can be computed by the nearest sonar unit, and the approximate direction also can be determined from the relative location of the sonar unit, which detects the nearest distance. The robot can be efficiently programmed to implement obstacle avoidance. However, to additionally implement person-tracking task, even in an environment with a fixed condition, is still difficult and not practical.

3. Transmitter-and-receiver Based Approach

Using a transmitter-and-receiver approach, the transmitters located on the target person transmit signals, such as ultrasonic waves or blinking LED. The receivers located on the robot receive those signals. After computing the distance and the angle of the target person from those signals, the robot knows where to move in order to turn itself toward the target person and decrease the distance in between. In [16], two transmitter-and-receiver based approaches have been discussed.

a. Person Tracking Using Blinking LED Devices

This approach requires equipping the target person with two infrared LED devices with fixed distance between them and using a camera on the robot to detect the two devices. This is similar to the vision-based approach. The main difference is that the signals from infrared LED devices should be firmer and not affected by the disturbance in the environment, as long as they are not blocked by any obstacle. The camera is able to rotate to keep the target person in the middle of the image. By computing the distance between two LED lights and the deviation of the two lights from the central vertical axis in the image, the range and the bearing of the target person can be obtained respectively by the robot.

b. Person Tracking Using an Ultrasonic Positioning System

This approach is to equip the ultrasonic transmitters on the target person and the receivers on the robot. By computing the time interval between transmitting and receiving the ultrasonic signal, the distance between the target person and the robot will be determined. The angle can also be computed from the time delay between several receivers.

These approaches are straight-forward for person-tracking, but they are not suitable when there are obstacles between the target person and the robot. The detection of obstacles will be a problem using these approaches. Without any additional mechanism, the robot is not able to implement obstacle avoidance.

4. Intelligent Space Approach

The intelligent space approach [17,18] utilizes several sensors, such as visual or non-visual sensors that are located in the environment to detect both the robot and the target person. Therefore, the position information of the robot and target person will be in the global coordinate and determined by the sensors in the intelligent space. From the relative positions of the robot and the target person, the robot motion will be planned by this intelligent space and controlled through the network. However, the desired approach in this research is to design an autonomous robot that implements tasks in unstructured environments. This approach then becomes unsuitable although it may be well-functioned.

5. Combined/Multi-Modal Approach

A combined/multi-modal approach [19] is made of a combination of several approaches. It is able to gather the advantages of each single approach. This is also the key subject in this thesis. By using an ultrasonic positioning system along with the sonar rangefinders, this research combines the transmitter-and-receiver based approach with the non-vision based approach. A suitable algorithm also will be designed to adapt the robot to several situations that may happen in the implementation of person-tracking. The robot can then accomplish the person-tracking tasks, which include person-following and obstacle avoidance in unstructured environments. Figure 1 shows the person-tracking mobile robot using an ultrasonic positioning system. Figure 2 shows the specially made vest equipped with ultrasonic transmitters.

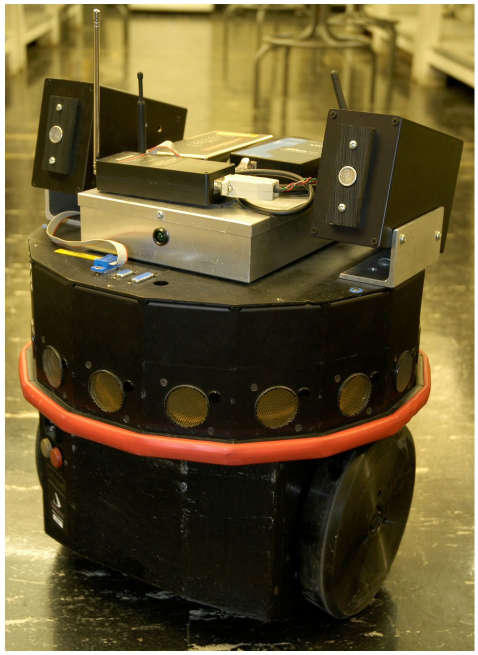

Figure 1. Person-Tracking Mobile Robot using an Ultrasonic Positioning System.

Figure 2. Specially Made Vest and Ultrasonic Transmitter.

D. THESIS OBJECTIVES

The main idea in this thesis is to investigate the feasibility of developing a person-tracking robot system using an ultrasonic positioning system, besides the 16 sonar sensors equipped on the Nomad 200 mobile robot. Furthermore, it will be proven to be the most reliable way to create a person-tracking mobile robot after completing the following steps.

1. Create the interface between the ultrasonic positioning system and the robot system in the operating program.
2. Design an algorithm that is able to simultaneously avoid obstacles and track the designated person in an unstructured environment.
3. Complete the task of person-tracking when there is no obstacle between the robot and the target person.
4. Complete the task of person-tracking when there is an obstacle between the robot and the target person.
5. Complete the task of person-tracking when the target person makes a turn at a corner.
6. Complete the task of person-tracking in an unstructured environment.

E. THESIS OUTLINE

A basic conception of the person-tracking mobile robot and the objectives of this thesis that will make a breakthrough in literature are summarized in Chapter I. The system architecture, including the hardware and the system configuration is described in Chapter II. The complete algorithm of the person-tracking mobile robot is illustrated in Chapter III. Several scenarios used to examine the motion of the robot and the results are described in Chapter IV. This thesis concludes in Chapter V. The C++ code used to implement the overall function of the robot is attached in the Appendix.

II. SYSTEM ARCHITECTURE

A. ULTRASONIC POSITIONING SYSTEM

The ultrasonic positioning system used in this research was made by SJAutomation L.L.C. The system was originally used to perform target tracking as a fixed, sensor system. By equipping two ultrasonic receivers at two fixed points in the environment, the system can continuously track the position of the target, which is equipped with the ultrasonic transmitters. The precision of target tracking is better if the distance between the two receivers is longer, as long as the receivers are in the effective range of the ultrasonic waves. In this research, the functions of the system are similar. The difference is that the system is mounted on the robot, so that the locations of the receivers are no longer fixed, but vary continuously while the robot is moving. Only the distance between two receivers is fixed. The ultrasonic positioning system can compute the relative position of the target from the robot in the form of range and bearing, in inches and degrees respectively.

Besides the ultrasonic transmitters and receivers, the ultrasonic positioning system includes also the RF (Radio Frequency) transmitter and receiver. The RF transmitter and receiver are mounted near the ultrasonic receivers and transmitters, respectively, as shown in Figure 3. The RF transmitter sends an electromagnetic signal to the RF receiver to request the ultrasonic signals from the ultrasonic transmitters. As soon as the RF receiver gets the RF signal, the ultrasonic transmitters transmit ultrasonic signals. Meanwhile, the ultrasonic receivers start to wait for the ultrasonic signals. The RF signal travels at the speed of light. As a result, the time spent for the RF signal to travel from the RF transmitter to the RF receiver is relatively short and can be neglected.

The algorithm of the ultrasonic positioning system, corresponding to Figure 3, is to compute the distance, D^*, between the target and the center of the two ultrasonic receivers, and the target bearing, γ. The distance between the ultrasonic receivers, d, is a fixed value. Assume that the time intervals for Receiver B and Receiver A to receive the signal transmitted from the transmitter are t_1 and t_2 respectively. The distances between both receivers and the transmitter can then be obtained by

$$D_1 = v t_1 \tag{2.1}$$

$$D_2 = v t_2 \tag{2.2}$$

where

$$v = Speed\ of\ Sound. \tag{2.3}$$

As a result, D_1, D_2, and d, are regarded as known. From the side-angle relations, the parameters in Figure 3 can be computed as follows:

$$\alpha = \cos^{-1}\left(\frac{d^2 + D_2^2 - D_1^2}{2dD_2}\right) \tag{2.4}$$

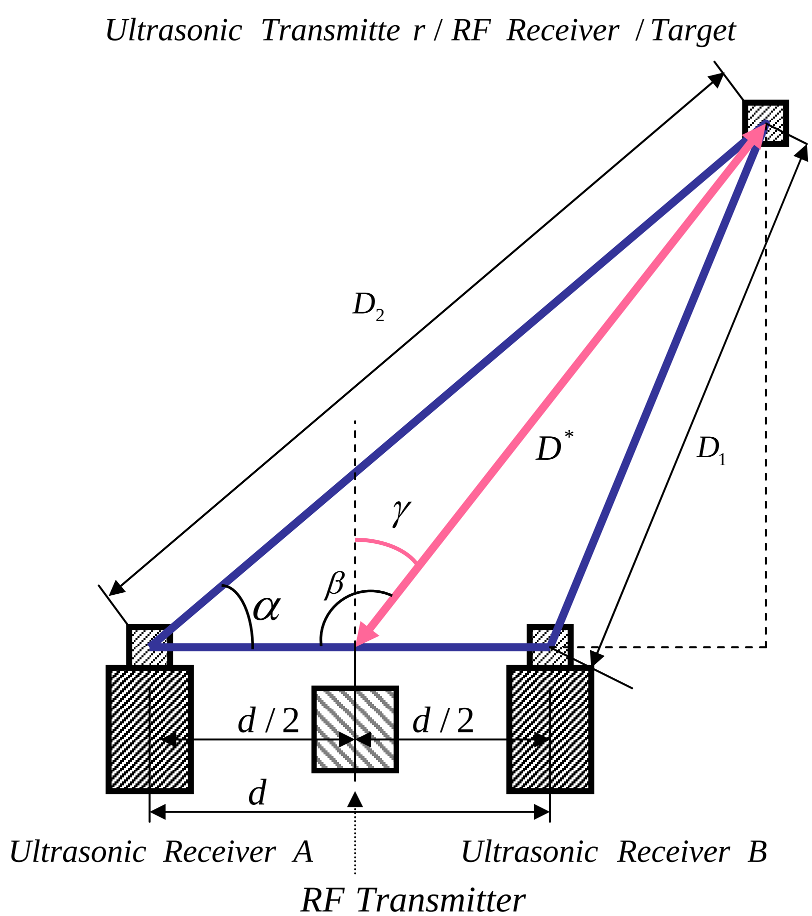

Figure 3. Ultrasonic Positioning System.

$$D^* = \sqrt{D_2^2 + (\frac{d}{2})^2 - 2D_2 \frac{d}{2} \cos\alpha} \tag{2.5}$$

$$\beta = \cos^{-1}\left(\frac{\left(D^*\right)^2 + \left(\frac{d}{2}\right)^2 - D_2^2}{2\left(\frac{d}{2}\right)D^*} \right). \tag{2.6}$$

The ultrasonic positioning system is configured to measure the angle of the target between -90° and 90°. The value will be positive when the target is on the left-hand side of the central vertical axis, and negative on the right-hand side. Therefore, the angle, γ, is obtained by

$$\gamma = 90^\circ - \beta. \tag{2.7}$$

By combining Equation 2.4 with Equation 2.5, the target distance, D^*, can be computed in the following equation.

$$D^* = \sqrt{\frac{2D_1^2 + 2D_2^2 - d^2}{4}}. \tag{2.8}$$

By combining Equation 2.6, Equation 2.7, and Equation 2.8, the target bearing, γ, can be computed by

$$\gamma = 90^\circ - \cos^{-1}\left(\frac{D_1^2 - D_2^2}{d\sqrt{2D_1^2 + 2D_2^2 - d^2}} \right). \tag{2.9}$$

The maximum distance, which can be measured from the target, depends on the maximum length of the time interval, which allows the receivers to wait for the signal to arrive. After this interval, the receivers will no longer receive signals until they are triggered again for the next cycle. The maximum time interval is called the "window." The maximum window size in this research has been configured as 20 milliseconds, in which the ultrasonic wave can travel 270 inches in room temperature, 20° C, approximately. When the signal arrives in less than 20 milliseconds, the system will shut down the window immediately after the first signal has been received. Otherwise, the system will only wait for 20 milliseconds and simply close the window right away, whether the signal has been received or not. When no signal is received, the values of the distance and the angle are not updated and remain the same as the last values.

B. ROBOT SYSTEM

The Nomad 200 mobile robot was made by Nomadic Technologies, Inc. This kind of robot uses a multiprocessor as a low level control system to control the sensing, communications, and motors. A remote workstation with Linux operating system is used as a high level control system to communicate with the robot multiprocessor and the ultrasonic positioning system through the wireless network. A laptop mounted on the robot can also be used to substitute the remote workstation.

The robot system is controlled using the C/C++ programming language. The information about the sensor systems and the motor state are stored in a global array, called "State Vector" [20, 21]. The reference to the states is shown in Table 1. In this section, several sensor systems equipped on the robot will be explained.

	Name	State Vector
0	STATE_SIM_SPEED	Speed of Simulation
…	…	…
17	STATE_SONAR_0	Sonar Data #0
18	STATE_SONAR_1	Sonar Data #1
19	STATE_SONAR_2	Sonar Data #2
…	…	…
32	STATE_SONAR_15	Sonar Data #15
33	STATE_BUMPER	Bumper Data
34	STATE_CONF_X	X Position
35	STATE_CONF_Y	Y Position
36	STATE_CONF_STEER	Steering Angle
…	…	…
38	STATE_VEL_RIGHT	Velocity of the Right Wheel
39	STATE_VEL_LEFT	Velocity of the Left Wheel
…	…	…
41	STATE_MOTOR_STATUS	Motor Status
44	STATE_ERROR	Error Number

Table 1. The State Vectors of the Robot System.

1. Bumper Sensors

The bumper sensors provide a mechanism that can be used to prevent damage to the robot motors when the robot runs into something. There are six individual bumper sensors arranged in a ring located on the robot. The n^{th} bumper sensor represents the n^{th} bit in the STATE_BUMPER vector, while the 0^{th} bit is the least significant one. A bit of the vector is set to 1 when the corresponding bumper is hit. In the robot algorithm, the robot simply stops when any of those bumpers is hit; that is, when the STATE_BUMPER vector is greater than 0. In Figure 4, the arrangement of the bumper sensors is presented.

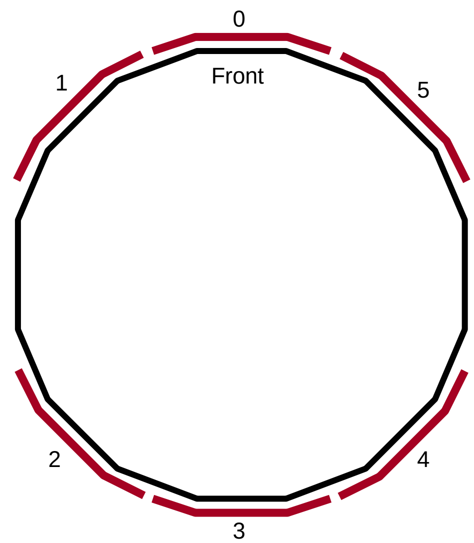

Figure 4. The Arrangement of the Bumper Sensors.

2. Sonar Sensors

There are 16 sonar units arranged in a ring located on the robot. In Figure 5, the arrangement of the sonar sensors is presented. The sonar units are numbered in counter-clockwise order beginning with the front of the robot. They emit sonar waves and receive echoes consecutively in this order, with a blanking period between each cycle of emitting and receiving. Note that the blanking period starts after the end of the processing of the previous sonar sensor, and ends before the beginning of the next one. The fire rate of the sonar sensors can then be adjusted by varying this period. In this research, the blanking period has been configured to be 50ms. From the time interval between the transmission of the sonar wave and the receiving of the echo, the distance between the robot and obstacles can be determined. The sonar sensors can measure distances from 6 inches to

255 inches. If an echo is not received, the sensor will regard the distance as 255 inches. The distance information will be stored in the state vectors, STATE_SONAR_0 to STATE_SONAR_15.

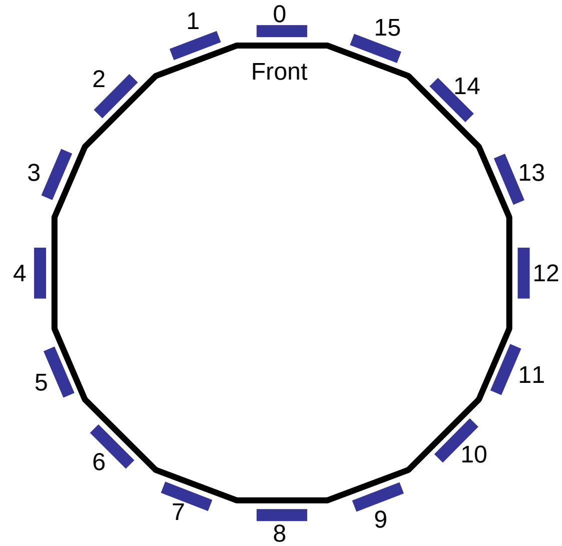

Figure 5. The Arrangement of the Sonar Sensors.

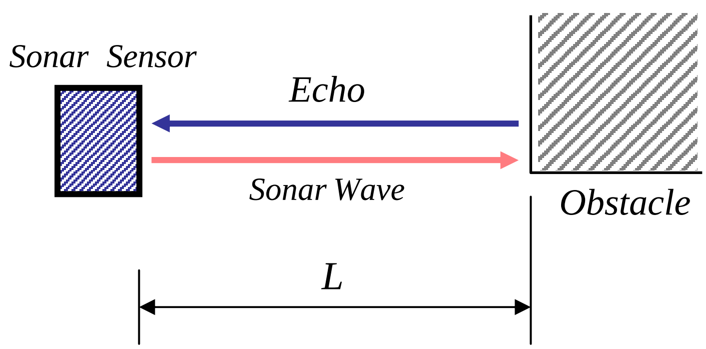

Figure 6. The Processing of a Sonar Sensor.

The method used to compute the distance between the robot and the obstacle by the sonar sensor is relatively more straightforward, as compared with the algorithm used in the ultrasonic positioning system. Figure 6 shows a scenario in which the distance, L, needs to be computed. Assume the time spent for the sonar sensor to receive the echo after the transmission of the sonar wave is T, which is a round-trip time period. The distance, L, is obtained by the following equation.

$$L = vT/2. \tag{2.10}$$

The parameter, v, is the sound speed.

3. Motor/Motion Sensors

The motor/motion information is updated consecutively in state vectors, 34 to 41. The values of STATE_CONFIG_X and STATE_CONFIG_Y are the integrated x-coordinate and y-coordinate positions, which are in 1/10 of inches, with respect to the start positions. The value of STATE_CONF_STEER is the robot steering angle in 1/10 of degrees, with respect to the start orientation. It is in the range, $[0;3600)$. As for the vectors, STATE_VEL_RIGHT and STATE_VEL_LEFT are the velocity of the right and left wheels in 1/10 of inches per second. The state vector of the motors, that is, STATE_MOTOR_STATUS, presents the statuses of the motors. Figure 7 shows the details of the status values.

• • •		ES	CH	AC	B1	B0	L	R
	7	6	5	4	3	2	1	0

R : set when the right wheel is in motion L : set when the left wheel is in motion

AC : set when the Scout is plugged into an AC source

CH : set when the Scout is plugged into an AC source and the batteries are charging

B1, B0 : 0, 0 Low Battery 0, 1 Med Battery 1, 0 High Battery 1, 1 Reserved

ES : set when the Emergency-Stop is down (always 0 for robots without E-Stop)

Figure 7. The Values of the Motor State Vector Bitmap.

C. INTERACTION BETWEEN ULTRASONIC POSITIONING SYSTEM AND SONAR SENSORS

The ultrasonic positioning system and the sonar sensors were previously two independent systems with their own periods. Their periods can be adjusted as short as possible to speed up the reaction of the robot. For example, the blanking period between two processes of sonar sensors can be adjusted to as little as 2 milliseconds. However, according to the experiments, the interference of both acoustic waves between the two sensor systems can be so significant that the readings of distances and angles become irrationally large. The robot may finally run out of control and probably crash into something. The best way to resolve this issue is to prevent the processes of the two systems from overlapping. An approach similar to the asynchronous handshake is used. Figure 8 shows the process timing diagrams of the ultrasonic and sonar devices. Table 2 shows the values of the timing parameters in Figure 8.

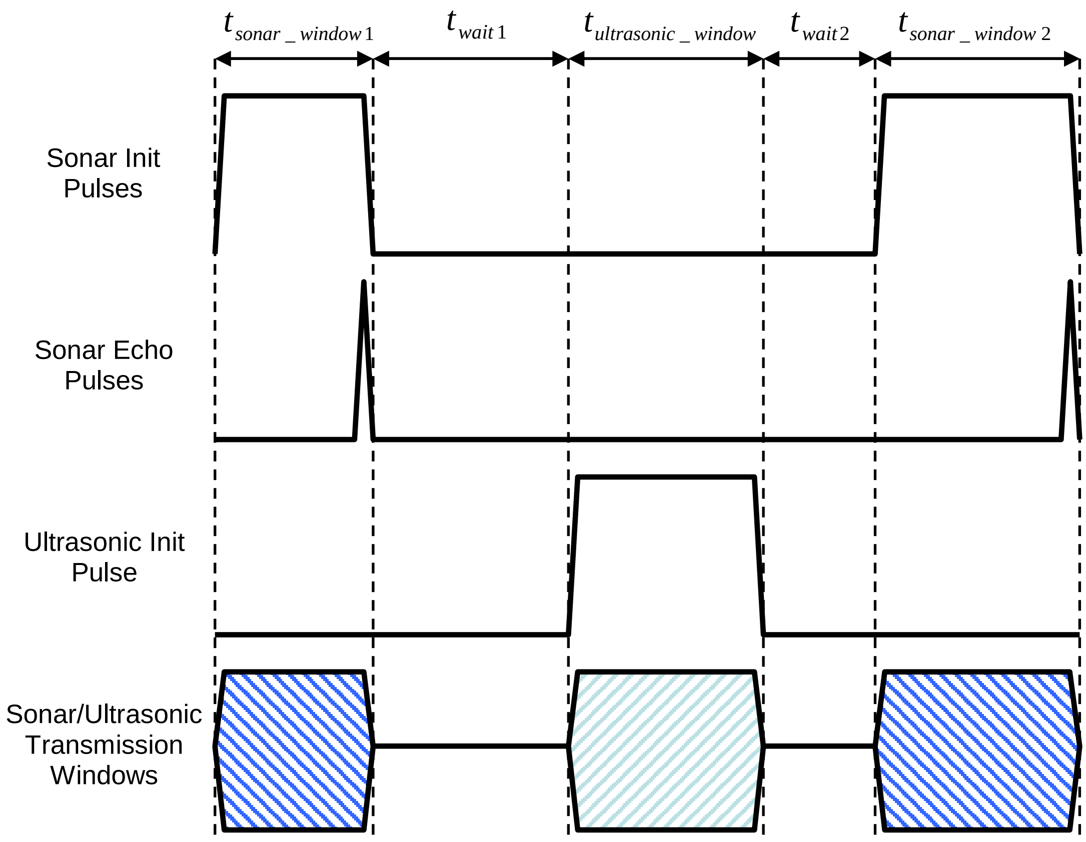

Figure 8. Process Timing Diagrams of Sonar and Ultrasonic Devices.

Parameter Name	Symbol	Value
Sonar transmission window 1	$t_{sonar\_window\,1}$	Time needed to receive the echo after sonar wave is transmitted (20 ms maximum)
Sonar transmission window 2	$t_{sonar\_window\,2}$	Time needed to receive the echo after sonar wave is transmitted (20 ms maximum)
Sonar blanking period	$t_{wait\,1} + t_{ultrasonic\_window} + t_{wait\,2}$	50 ms
Waiting period before the transmission of the ultrasonic wave	$t_{wait\,1}$	20 ms
Ultrasonic wave transmission window	$t_{ultrasonic\_window}$	20 ms maximum
Waiting period After the transmission of the ultrasonic wave	$t_{wait\,2}$	50 ms - $t_{wait\,1}$ - $t_{ultrasonic\_window}$

Table 2. Timing Parameters.

The beginning of the sonar init pulse starts when the sonar wave is being transmitted, and this pulse is terminated by the rising edge of the echo pulse; that is, the sonar transmission window will be closed as soon as the echo is received. As for the transmission of the ultrasonic wave, after the falling edge of the sonar init pulse, the ultrasonic positioning system waits 20 milliseconds for the sonar waves to die out, and immediately transmits a kind of electromagnetic signal from the RF transmitter to the RF receiver on the target to request ultrasonic signals. The time needed for the electromagnetic signal to travel to the target is relatively short and can be neglected. Simultaneously, the ultrasonic positioning system opens a window with a maximum length of 20 milliseconds for the ultrasonic wave (transmitted from the ultrasonic transmitters) to arrive. Therefore, there will be no overlap between the processes of both sonar sensor and the ultrasonic sensor. The interference problem can be avoided.

Because of the longer time interval that is configured between the processes of any two sonar sensors, the time needed for all 16 sonar sensors to complete their processes once, will be relatively longer. As a result, the speed to update the information of the obstacle distances will not be fast enough for the robot to react. An effective approach to deal with the speed issue is to enable only the front 5 sonar sensors - since only sonar sensors in the front of the robot are needed for this forward-motion-only implementation, and since there is no significant influence to turn off those other sonar sensors in the back during the implementation. The time needed for the same sonar sensor to transmit again will be sufficiently shortened. Therefore, the person-tracking implementation can obtain a better result.

D. SUMMARY

The system architecture including sonar sensor system, motor system, bumper sensor system, and ultrasonic positioning system are presented in this chapter. In order to adapt the ultrasonic positioning system to the original robot system, the interference between the sonar sensor system and the ultrasonic positioning system should be avoided. An asynchronous handshake method is adopted to schedule the actions of these two systems that address this situation. Furthermore, to accelerate the reaction of the robot, a limited number of the sonar sensors, which are indispensable for most person-tracking conditions, are used in the implementation.

THIS PAGE INTENTIONALLY LEFT BLANK

III. ALGORITHM

A. POTENTIAL FIELD ALGORITHM

1. Robot Coordinate

The coordinate used in the potential field algorithm is the robot coordinate. Since the robot is moving on a flat surface, there are only two dimensions. The origin is the robot center. The two axes, *x-axis* and *y-axis*, are toward the front and the left of the robot, respectively. The diagram for the robot coordinate is shown in Figure 9.

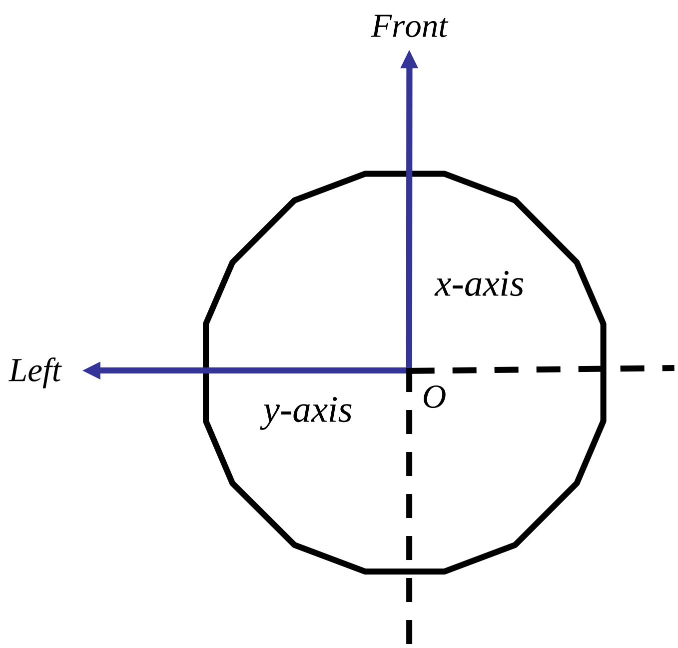

Figure 9. Robot Coordinate.

2. Attractive Forces Derived From the Readings of the Ultrasonic Sensor

The attractive forces in the two axes of the robot coordinate are computed from the distance, D^*, and the bearing, γ. Figure 10 shows an example presenting these two attractive forces, $F_{ultrasonic\_x}$ and $F_{ultrasonic\_y}$, in *x-axis* and *y-axis* respectively. They are computed as follows:

$$F_{ultrasonic\_x} = D^* \cos(\gamma) \tag{3.1}$$

$$F_{ultrasonic\_y} = D^* \sin(\gamma). \tag{3.2}$$

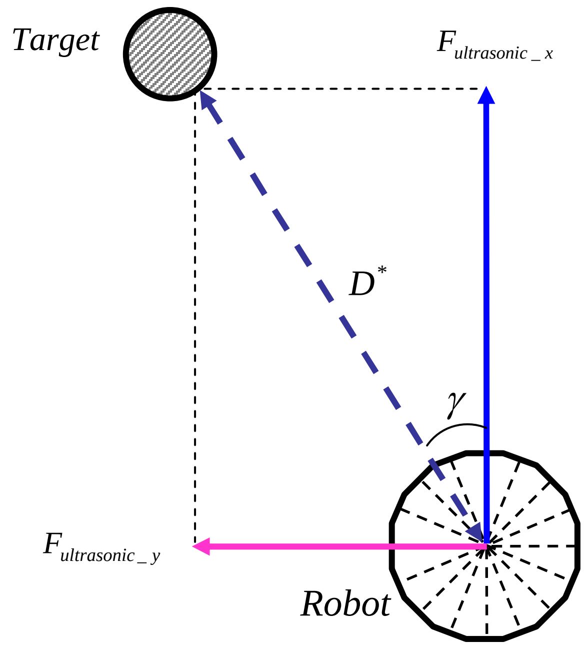

Figure 10. Attractive Forces Derived From the Readings of the Ultrasonic Sensor.

3. Attractive Forces Derived From the Readings of Sonar Sensors

The simplified diagram of the sonar sensor readings is shown in Figure 11. In the potential field algorithm, the distance values measured by the sonar sensors are regarded as the attractive forces. According to several experiments, assuming the robot is always facing the target and all the sonar sensors are in use, the combined force in *x-axis* may cause severe oscillations in the robot motion when the robot is close to the target. This occurs because of the following situation. When the robot is near the target, the distance readings of the sonar sensors in the front of the robot are smaller than those in the back. The combined force derived from these readings forces the robot to go backward. As soon as the robot moves backward, the attractive force derived from the readings of the ultrasonic sensor increases and forces the robot to go forward, resulting from the increase in the distance between the robot and the target. In addition, there are processing and network delays in the implementation. It will not be easy for the robot to settle down to a balance point because of those delays. Therefore, it is not necessary to adopt this part of the attractive force, which introduces oscillations. To deal with this issue, only the weights in *y-axis* of the attractive forces are used in computing the combined force.

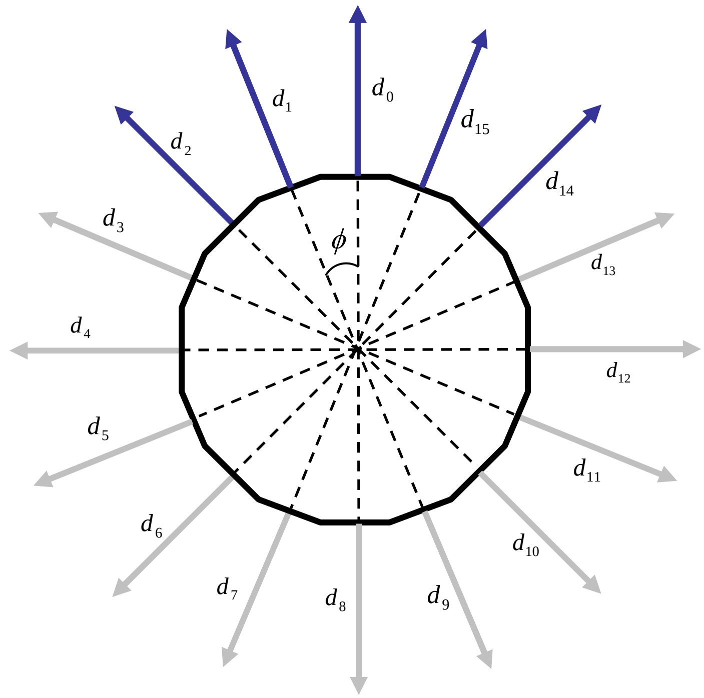

Figure 11. Distances Measured by Sonar Sensors.

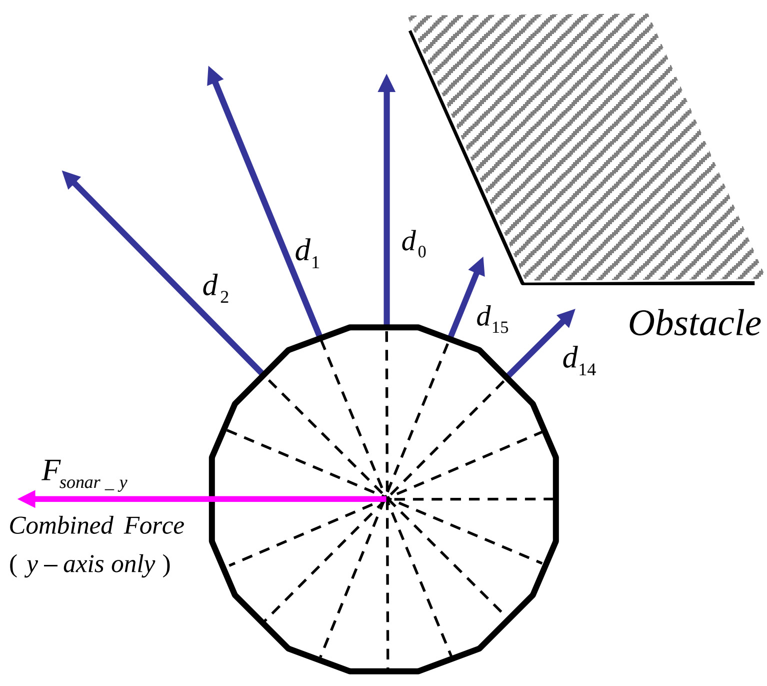

Figure 12. Combined Force Derived From the Readings of the Sonar Sensors.

Therefore, the combined force in *x-axis*,

$$F_{sonar\_x} = 0. \tag{3.3}$$

The combined force in *y-axis* can be computed as follows:

$$F_{sonar\_y} = \sum_{n=0}^{15} d_n \sin(n\phi)$$

where $\phi = 22.5^\circ$. Since only five of the sonar sensors in the front of the robot are used in the implementation, the equation becomes

$$F_{sonar\_y} = d_2 \sin(2\phi) + d_1 \sin(1\phi) + d_{15} \sin(15\phi) + d_{14} \sin(14\phi). \tag{3.4}$$

Figure 12 shows how the attractive forces will react when the robot encounters an obstacle. If the obstacle is on the right of the robot, the distance readings of the 14th and the 15th sonar sensors will be smaller. As a result, the combined attractive force will be toward the left.

4. Potential Field Motion Planning From Combined Forces

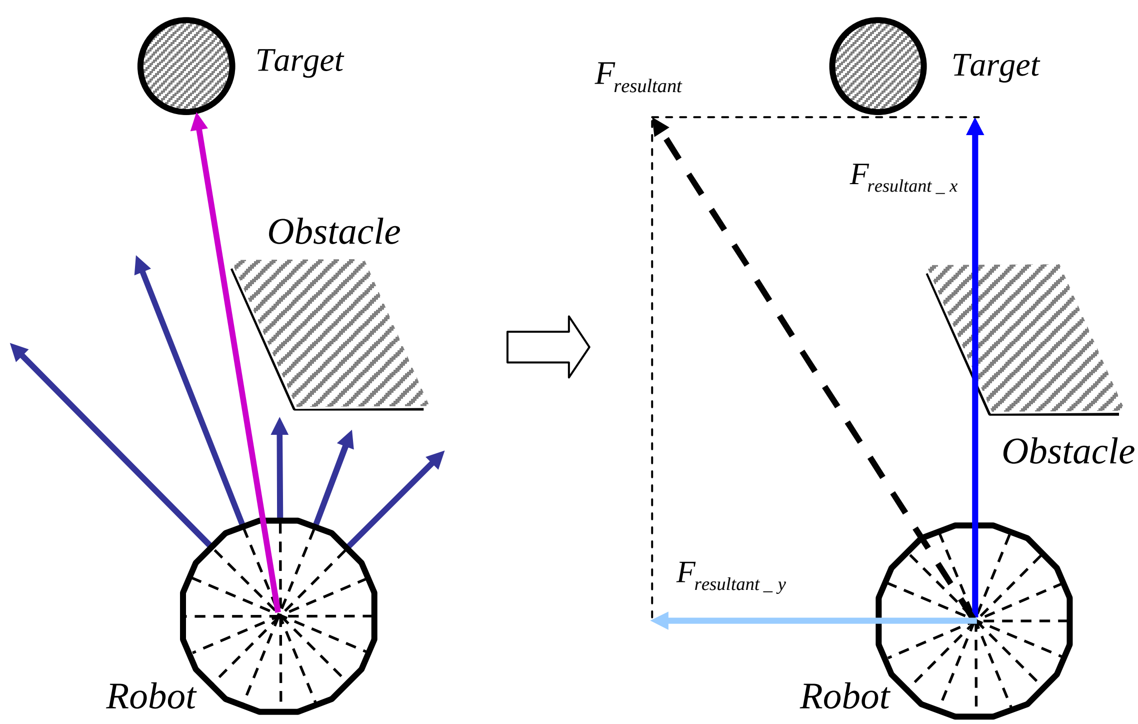

Figure 13. Resultant of the Attractive Forces.

The potential field motion planning is using the resultant of the attractive forces, which are derived from the readings of the ultrasonic positioning system and the sonar rangefinders. Figure 13 presents an example of the robot motion, which is corresponding to the resultant force, at exactly the moment of the relative position shown in the figure. The resultant force intends to drive the robot toward the target and away from the obstacle at the same time. It will be updated consecutively to control the motion of the robot in real time.

The resultant force can be computed as follows:

$$F_{resultant} = \begin{bmatrix} F_{resultant\_x} \\ F_{resultant\_y} \end{bmatrix} \tag{3.5}$$

$$F_{resultant\_x} = K_1 F_{ultrasonic\_x} + K_3 F_{sonar\_x} \tag{3.6}$$

$$F_{resultant\_y} = K_2 F_{ultrasonic\_y} + K_4 F_{sonar\_y}. \tag{3.7}$$

where K_1, K_2, K_3, and K_4 are adjustable and act as weighting parameters. Since the force, F_{sonar_x}, is equal to 0, the parameter, K_3, is meaningless and has no effect on the robot motion. In this research, those parameters are adopted according to numerous experiments, which include observing the behaviors of the robot in different situations and adjusting the values of the parameters. Those parameters are as follows:

$$\begin{cases} K_1 = 15 \\ K_2 = 20 \\ K_4 = 10. \end{cases} \tag{3.8}$$

An additional limit to the force, F_{sonar_y}, is used to prevent the combined force from the sonar sensors from becoming much larger than that from the ultrasonic sensor. That is, if $\left|F_{sonar\_y}\right| \geq K_5 \left|F_{ultrasonic\_y}\right|$, the equality, $\left|F_{sonar\_y}\right| = K_5 \left|F_{ultrasonic\_y}\right|$, will be used. The parameter, K_5, in this case is set to be 0.75.

5. Resultant Forces to Translation and Steering Velocities Conversion

In order to control the motion of the robot, the translation and steering velocities should be determined. In the potential field algorithm, the resultant forces, $F_{resultant\_x}$

and $F_{resultant\_y}$, are converted to translation velocity and steering velocity, respectively. They are obtained as follows:

$$V_{translation} = G_1 F_{resultant\_x} \quad ((1/10)\ \textit{inches/sec}) \tag{3.9}$$

$$V_{steering} = G_2 F_{resultant\_y} \quad ((1/10)\ \textit{degrees/sec}). \tag{3.10}$$

The values of the parameters, G_1 and G_2, are adjustable. In this research, those values are chosen as $G_1 = G_2 = 0.1$. In addition, a limit to the translation velocity is set to prevent the robot from moving too fast. That is, if $|V_{translation}| \geq 120\ ((1/10)\ inches/\sec)$, the equality, $|V_{translation}| = 120\ ((1/10)\ inches/\sec)$, will be used to limit the maximum absolute value of translation velocity to be $120\ ((1/10)\ inches/\sec)$. Those velocities, $V_{translation}$ and $V_{steering}$, are then the reference inputs to the robot motor system.

B. ALGORITHM USED AS THE TARGET IS IN A CERTAIN RANGE

When the potential field algorithm is used to implement person-tracking, it is necessary to add another mechanism to the overall algorithm, as the robot is close to the target in a certain range. Otherwise, the robot will keep approaching until it collides with the target. To deal with this, a certain range, which is a suitable distance between the robot and the target, should be determined. In this research, the range is adopted to be 65 inches. To prevent from the oscillation in the robot motion, this distance is not designed to be the shortest distance that has to be kept between the robot and the target. Instead, it is a signal from the distance reading of the ultrasonic positioning system to tell the robot to stop approaching the target. In other words, the robot will set its translation velocity, $V_{translation}$, to be zero when the target is within this range. The distance between the target and the robot could be less than this certain range once the robot actually stops, but no collision will occur in this case. The oscillation caused by the adjustment to keep a certain distance between the robot and the target, will not be an issue in this research.

Except for the termination of the translation velocity, the steering velocity of the robot is still active when the target is in the range of 65 inches. With this steering velocity, the robot simply turns itself to face the target without displacement in position. Again, to prevent oscillation while the robot is trying to adjust itself to keep the angle reading of the ultrasonic positioning system to be zero, an elastic range is determined to be from -10° to 10°. When the angle reading from the ultrasonic sensor is within this range, the robot will set its steering velocity, $V_{steering}$, to be zero. Therefore, the robot will be physically motionless when the range and bearing conditions are satisfied. When the target begins moving out of those ranges, the robot will activate itself again to implement all the relative movements in tracking the target.

When the target range, $D^* < 65$, the velocities, $V_{translation}$ and $V_{steering}$, in this algorithm can be determined as follows:

$$V_{translation} = 0 \tag{3.11}$$

$$V_{steering} = \begin{cases} G_3 F_{ultrasonic\_y}, & |\gamma| > 10 \\ 0, & |\gamma| \le 10. \end{cases} \tag{3.12}$$

The parameter, G_3, is an adjustable constant, and it is set to be 2 in the implementation. And the parameter, γ, is the target bearing obtained by the ultrasonic sensor.

C. OBSTACLE AVOIDANCE ALGORITHM

By using the potential field algorithm, the robot can already implement obstacle avoidance when the distance between the robot and the obstacle is not too short. However, the function required for obstacle avoidance is not fully complete. The two senor systems, ultrasonic sensor and sonar sensors, have to operate in coordination with each other. The adjustment in the values of the parameters in one situation may not be suitable in another. On the other hand, since an obstacle is always located between the robot and the target, the attractive force driving the robot toward the target can be partly regarded as the attractive force toward the obstacle. Therefore, it will still be possible for the robot to collide with an obstacle, when the attractive force driving the robot away from the obstacle is smaller than that driving the robot toward it. To correct this, an additional algorithm will be necessary when the robot is near the obstacles. Instead of using only the potential field algorithm, the robot system switches its algorithm to the one that is specialized in implementing obstacle avoidance. The potential field algorithm takes over the system as soon as the situation is resolved. The distance, which is defined to be the threshold between the robot and the obstacle, is a range of 12 inches. When the obstacle is within this range, the obstacle avoidance algorithm will be implemented. In this section, the motion planning will be discussed in three different conditions, which are as follows.

1. Motion Planning for Obstacles on the Right Forward of the Robot

In this section, the obstacle is on the right forward of the robot, corresponding to Figure 14. When either the 14th or the 15th sonar range is smaller than 12 inches, the robot will cease the translation velocity and make a left turn until those sonar readings are greater than or equal to 12. As soon as both the sonar ranges are larger than the 12 inch threshold, the obstacle avoidance algorithm will be terminated. The potential field algorithm will take over the system to carry on the implementation of person-tracking. When implementing obstacle avoidance, the steering velocity, $V_{steering}$, is an adjustable parameter and is chosen as

$$V_{steering} = 110 \quad ((1/10)\ degrees/sec) \tag{3.13}$$

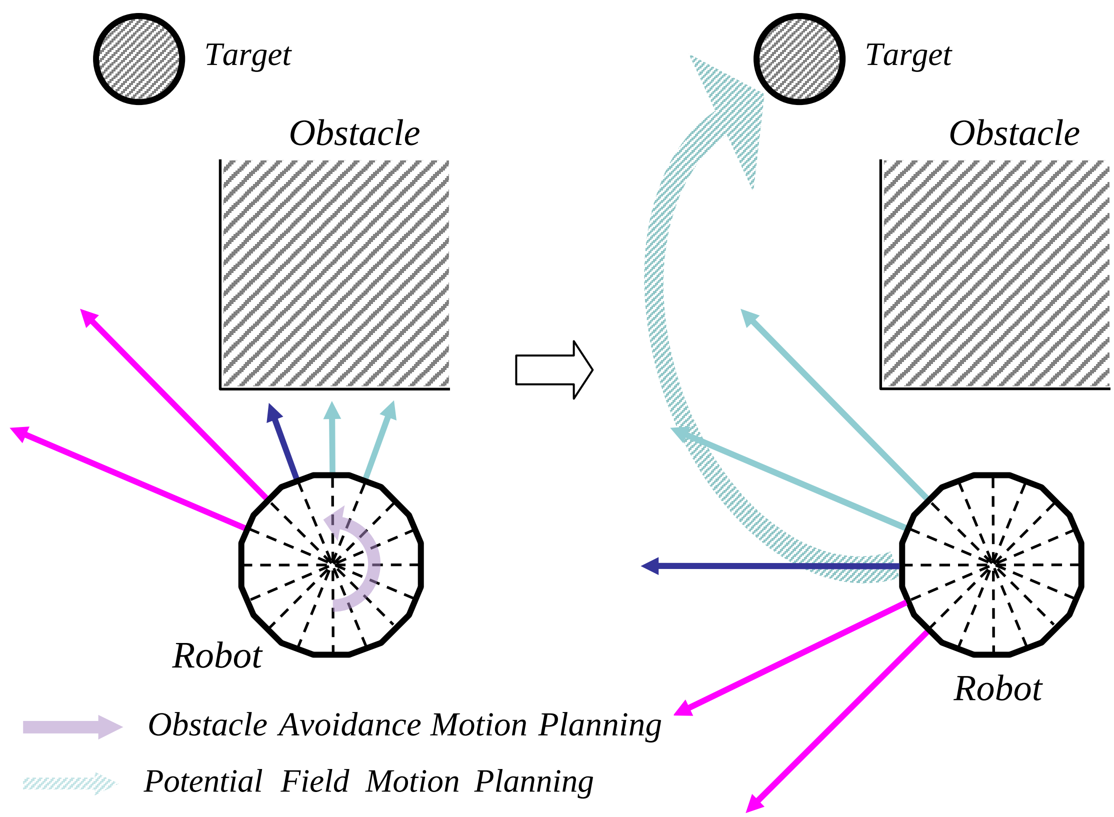

Figure 14. Motion Planning for Obstacles on the Right Forward of the Robot.

2. Motion Planning for Obstacles on the Left Forward of the Robot

When the obstacle is on the left forward of the robot, it is similar the obstacle being on the right forward, as in the previous section. The only difference is that the robot will make a right turn when it encounters the obstacle. That is when either the 1st or the 2nd sonar range is smaller than 12 inches. The potential field algorithm will take over the system as soon as this situation no longer exists. When implementing obstacle avoidance, the translation velocity, $V_{translation}$, is set to be 0, and the steering velocity, $V_{steering}$, is an adjustable parameter chosen as in Equation 3.14.

$$V_{steering} = -110 \quad ((1/10)\ degrees/sec). \tag{3.14}$$

The negative sign in Equation 3.14 is appended because the robot is making a right turn. On the other hand, the positive sign will be used when the robot is making a left turn.

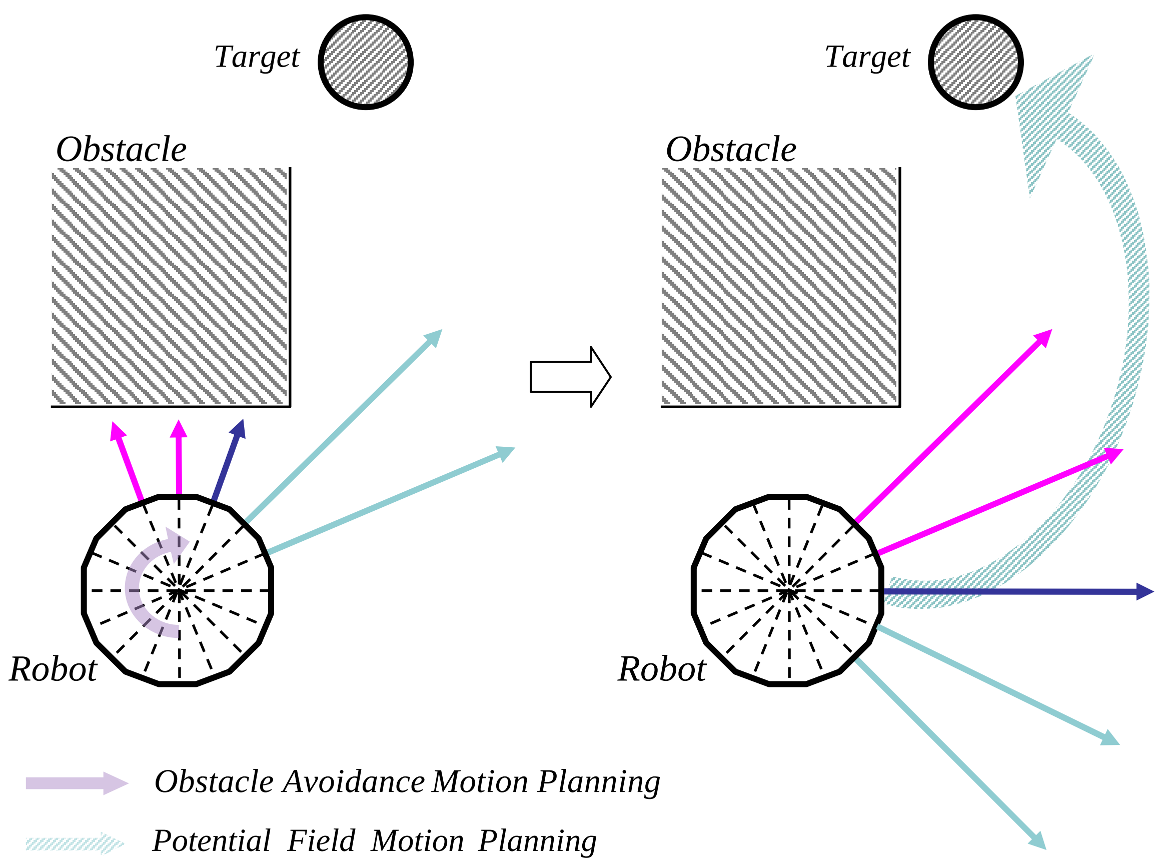

Figure 15. Motion Planning for Obstacles on the Left Forward of the Robot.

3. Motion Planning for Obstacles in Front of the Robot

When there is an obstacle between the target and the robot, the left side and the right side sonar ranges both may be smaller than the 12 inch threshold. The robot needs to decide which direction to turn in order to escape this problem. In this case, it is assumed that the obstacle is relatively small and does not obstruct the line of sight of the robot, such that the target transmitter signal can still be received. To deal with this, the target bearing is appended to the algorithm. When the target bearing, γ , is negative, as shown in Figure16, the robot will regard a right turn as a better decision to carry out the person-tracking task. On the other hand, the robot will make a left turn, when the target bearing, γ , is a value greater than or equal to zero, as shown in Figure 17. As soon as the robot turns itself away from the obstacle - when all the sonar ranges are no longer smaller than 12 inches - the potential field algorithm will take over the system again to carry on the person-tracking task.

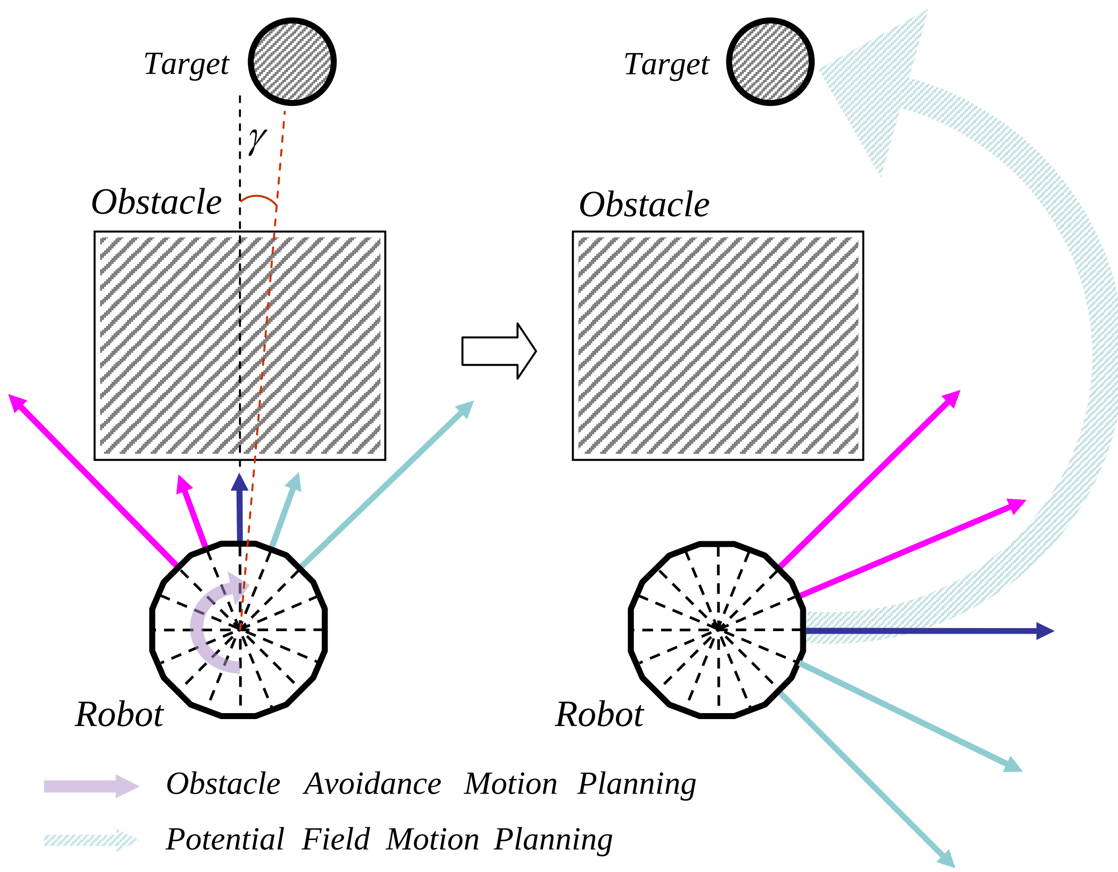

Figure 16. Motion Planning for Obstacles in Front of the Robot $(\gamma < 0)$.

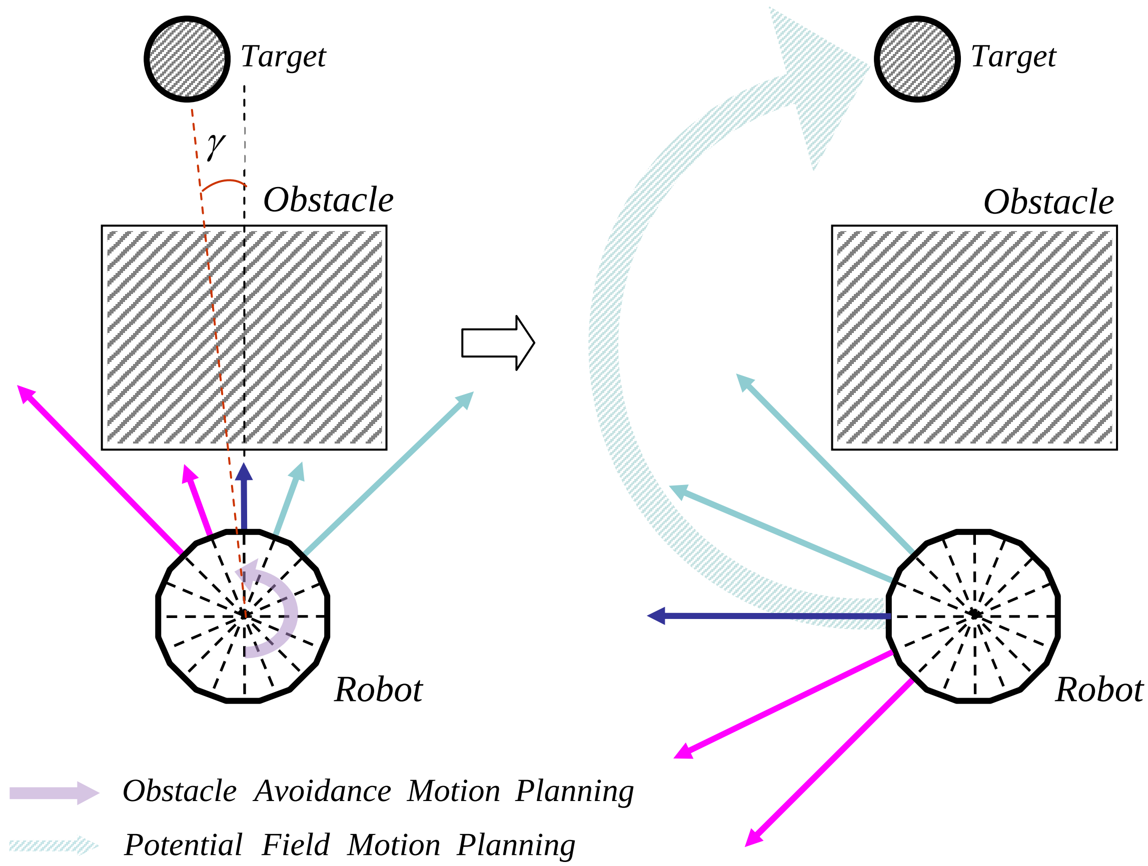

Figure 17. Motion Planning for Obstacles in Front of the Robot ($\gamma \geq 0$).

When implementing obstacle avoidance, the translation velocity, $V_{translation}$, is set to be 0, and the steering velocity, $V_{steering}$, is an adjustable parameter chosen as following equations.

$$\begin{cases} V_{steering} = 150, & \gamma \geq 0 \\ V_{steering} = -150, & \gamma < 0. \end{cases} \tag{3.15}$$

The parameter, γ, is the target bearing obtained directly from the readings of the ultrasonic positioning system.

Note that the motion planning methods used in these three situations are practically related to one another. For example, the third situation will be led to the first or the second situation, when the robot starts to make a turn, which immediately changes the sonar statuses.

D. OVERALL ALGORITHM OF A PERSON-TRACKING MOBILE ROBOT

By combining all the sub-algorithms described in this chapter, the simplified overall algorithm of a person-tracking mobile robot is shown in Figure 18. Note that there is prioritized order when carrying out the algorithm. The idea is to prevent the robot from collision, which could damage the robot system. Examining the bumper sensor status has high priority over others. When the STATE_BUMPER vector is set, the process should go directly to "stop." IF the STATE_BUMPER is not set, the main algorithm can be executed. The algorithm used when the target is in a certain range can be carried out before the obstacle avoidance algorithm, since there is no displacement in the robot's position when implementing this portion. Collision is not a concern in this case.

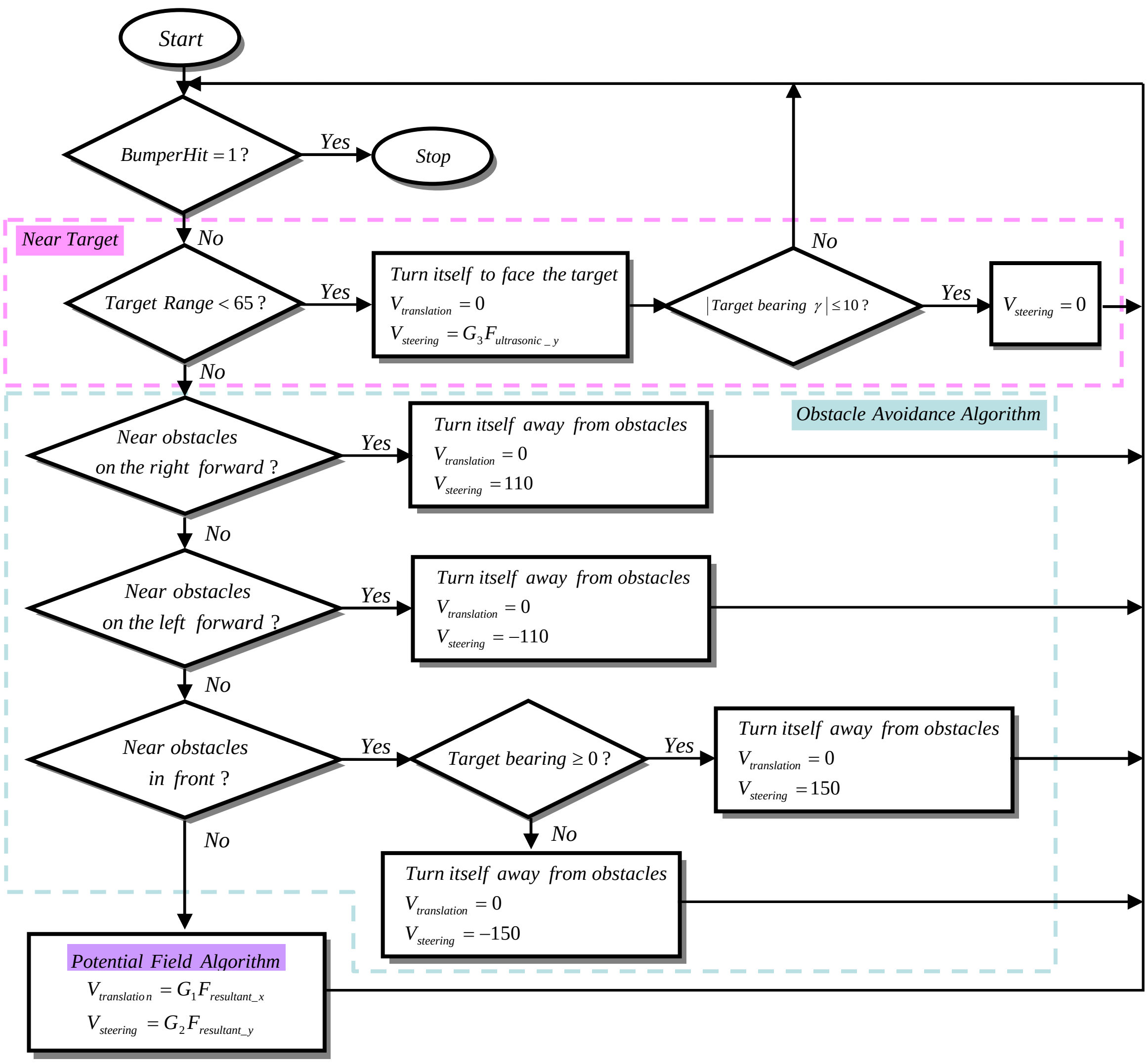

Figure 18. Overall Algorithm Flowchart.

When the target moves out of a 65 inch range, the system will examine the sonar states to determine if there are obstacles standing in the path. When the sonar ranges are smaller than the 12 inch threshold, the action for obstacle avoidance should be made. In the event that the former situation no longer exists, the potential field algorithm is carried out in tracking the target.

E. SUMMARY

In this chapter, the algorithms used to address several major motion control situations are illustrated in detail. The prioritized order to implement those algorithms is also described. One thing worth mentioning is that the termination of the robot program can be done either when the STATE_BUMPER vector is set or by terminating it directly on the remote workstation.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. EXPERIMENTS AND RESULTS

A. PERSON-TRACKING IMPLEMENTATION WITHOUT OBSTACLES

In this section, an experiment will be implemented to verify the ability of the person-tracking mobile robot in an obstacle-free situation. Figure 19 shows the procedure of this experiment. The robot attempts to follow when the target person is moving, and finally keeps itself within a certain range when the target person stops. Note that the robot trajectory can be recorded by the remote workstation. As shown in the figure, the trajectory is intentionally presented in several time scales, by which the interaction between the target person and the robot can be demonstrated conspicuously. According to the result of this fundamental test, as shown below, the completeness of the normal person-tracking ability performed by the robot is ensured. Therefore, other factors, such as obstacles and persons other than the target, can be included in further experiments. Those experiments with different situations will be discussed in the later sections.

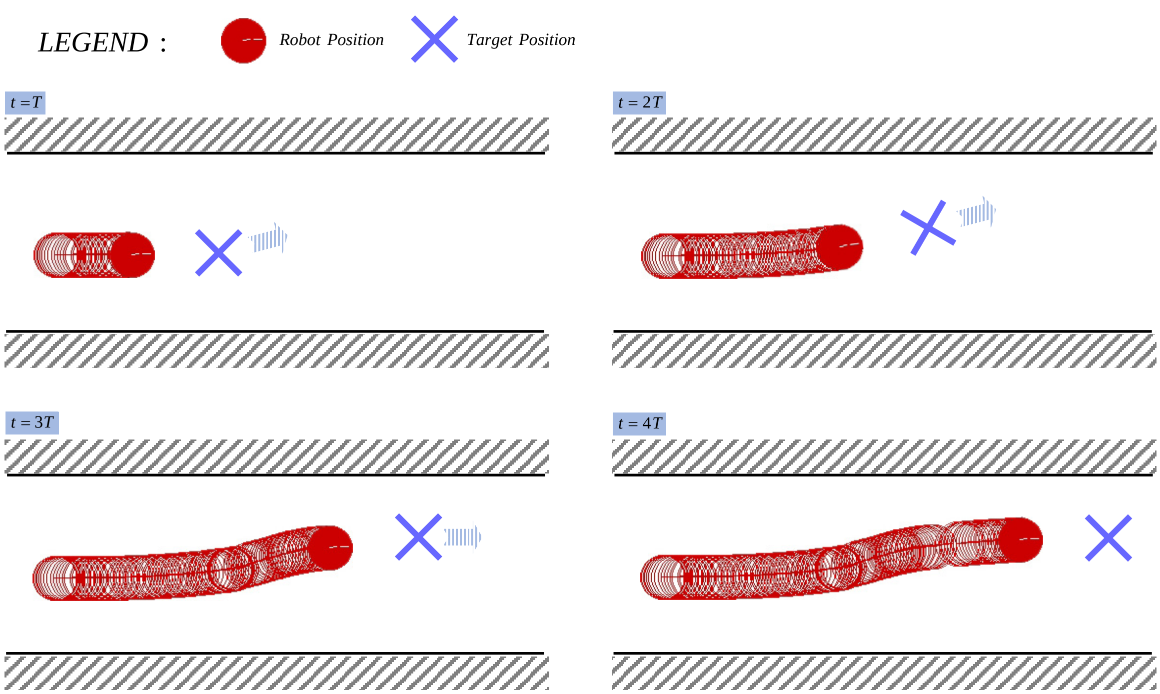

Figure 19. Robot Trajectory in an obstacle-free situation.

B. PERSON-TRACKING IMPLEMENTATION WITH AN OBSTACLE BETWEEN THE ROBOT AND THE TARGET PERSON

In this section, two main experiments will be carried out to verify the ability of the robot to implement obstacle avoidance during person-tracking. The first experiment corresponding to Figure 20 is to test the robot's behavior when the robot encounters an obstacle in the midst of person-tracking. The second experiment corresponding to Figure 21 is similar to the first one. The only difference is that the implementation of obstacle avoidance should be completed before the robot can carry out the person-tracking task. It is clear to see that the robot does have the ability to avoid the obstacle and accomplish the person-tracking task, which is actually difficult for most applications of the person-tracking robot in practice.

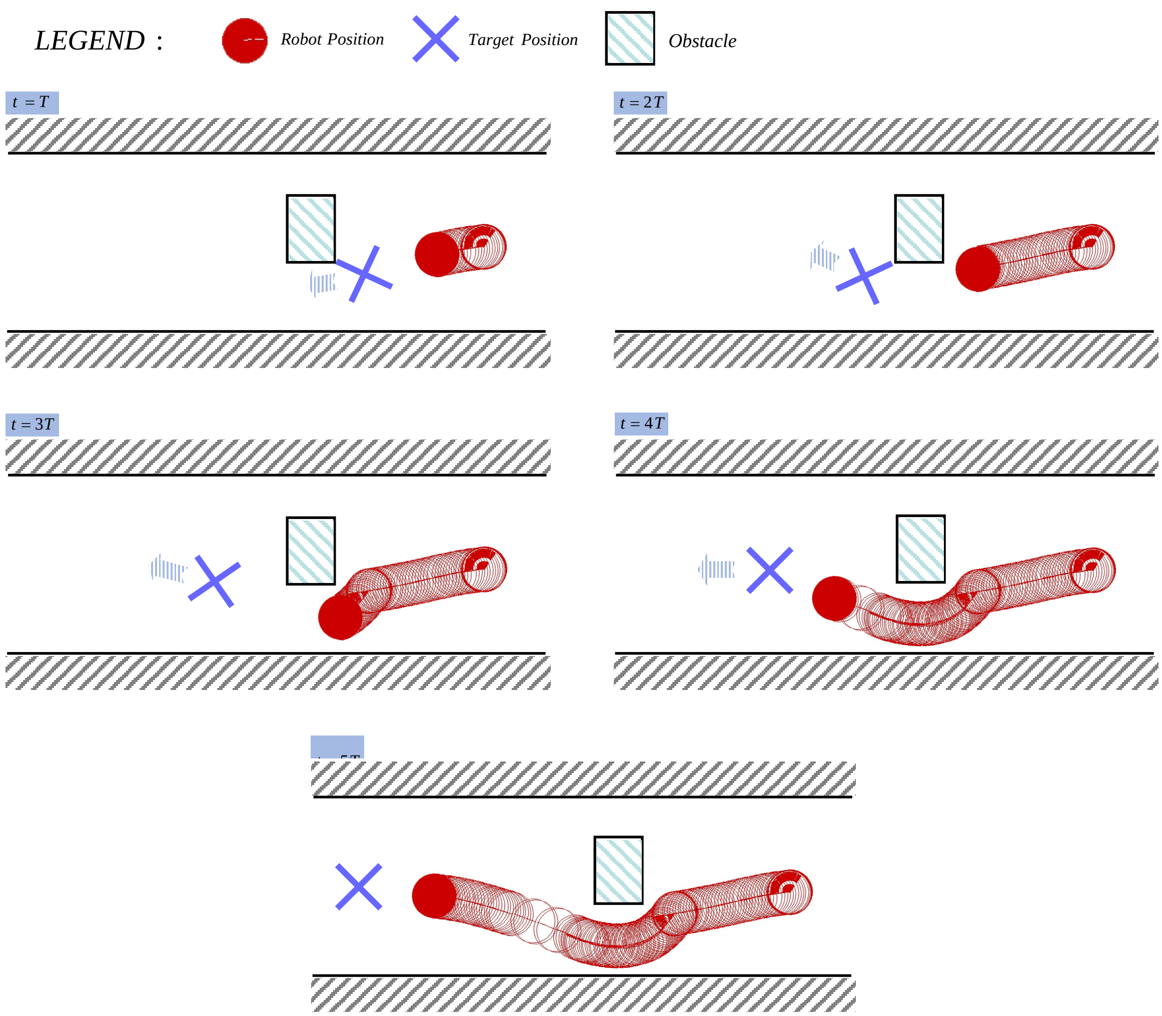

Figure 20. Robot Trajectory when encountering an obstacle.

Note that the obstacle is right in front of the robot in the second experiment corresponding to Figure 21. The robot should know which way to turn to avoid the obstacle simply by the target person's movement. As long as the target person moves away form the line of sight of the robot, the robot will turn according to the angle of the target person. The algorithm used has been described in Chapter III.

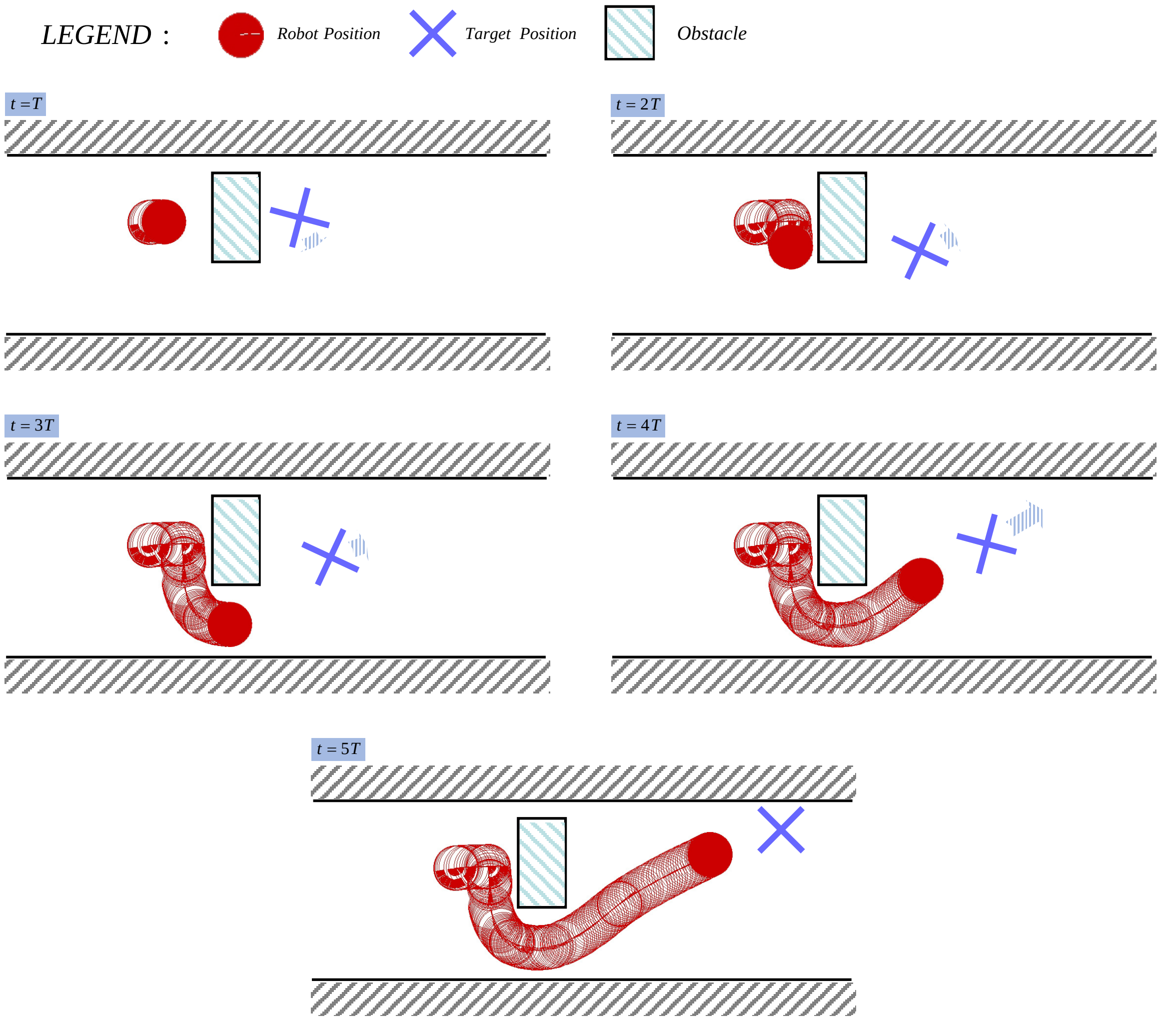

Figure 21. Robot Trajectory when there is an obstacle between the robot and the target person in the beginning of the person-tracking task.

C. PERSON-TRACKING IMPLEMENTATION WHEN THE TARGET PERSON MAKES A TURN AT A CORNER

In this section, an experiment will be implemented to verify the ability of the robot to carry out the person-tracking task when the target makes a turn at a corner. Figure 22 shows the person-tracking procedure and the robot trajectory.

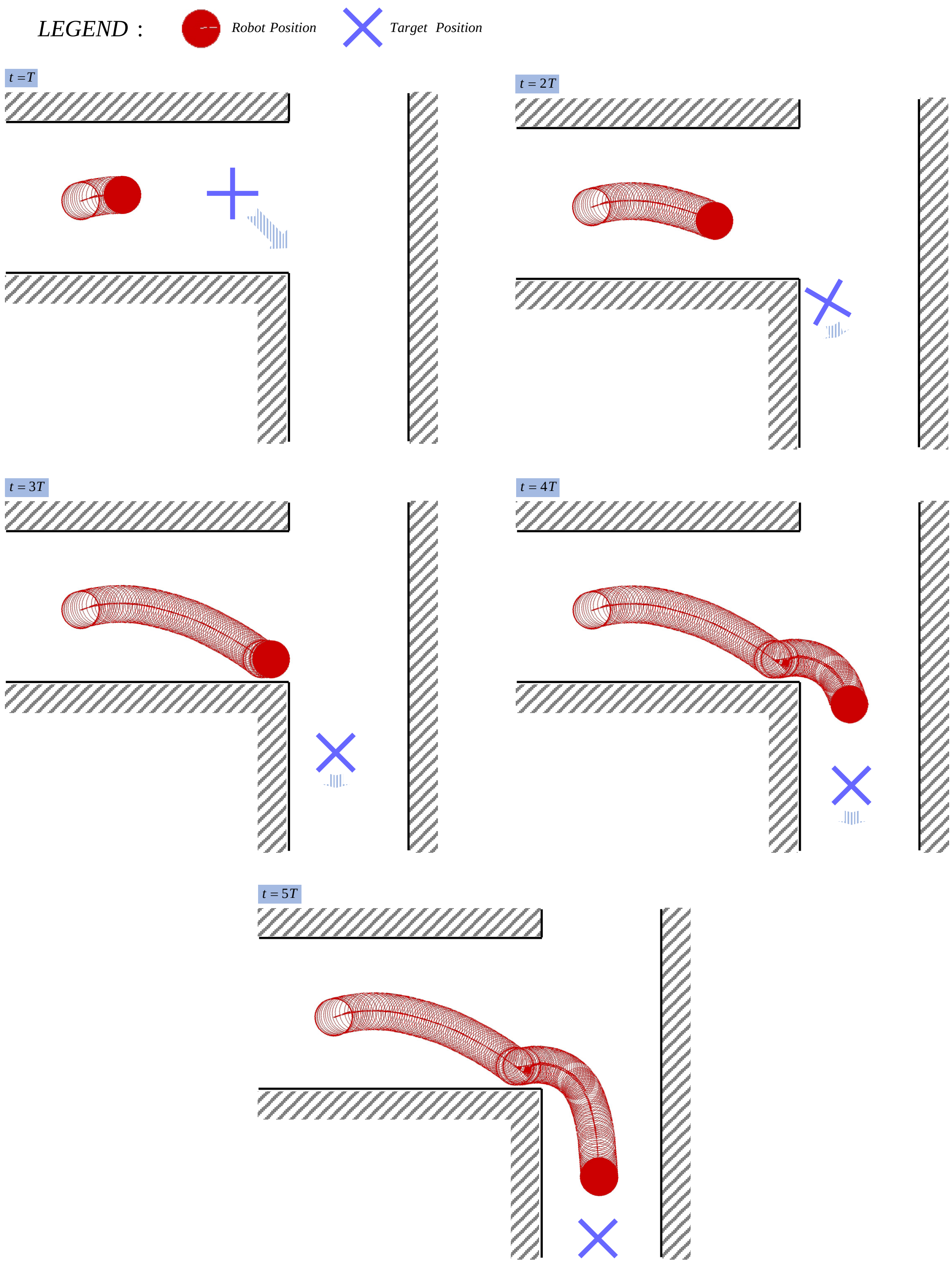

Figure 22. Robot Trajectory when the target person makes a turn at a corner.

Turning a corner is a common issue for most applications of the person-tracking robot. When the target person moves too fast at a corner, the corner will then become an obstacle. The robot will need to avoid the corner during the person-tracking task as shown in Figure 22. The feasibility of the robot's turning a corner has been proven during the implementation of this experiment.

D. PERSON-TRACKING IMPLEMENTATION IN AN UNSTRUCTURED ENVIRONMENT

Since several, sample situations have been coped with successfully, a more practical validation will be performed in this section. That is the person-tracking implementation in an unstructured environment. An unstructured environment has the following conditions. First, there will be relatively more obstacles in the environment. Second, the target person will not move along a certain fixed route. In other words, the route will be arbitrary. Third, there are persons, which may be unexpected, other than the target person wandering around in the environment. The objective of the examination is to verify the ability of the robot to carry out the person-tracking task in a more practical environment with unanticipated conditions.

Figure 23 shows the procedure of this examination and the robot trajectory. The robot performs the ability to implement obstacle-avoidance and person-following tasks during several turns and even through a narrow corridor. In addition, the robot is only following the target person and not affected by the movements of other persons in the environment. Note that the robot will regard the persons other than the target person as obstacles when they are too close in range. According to the robot trajectory in the figure, the robot following the wrong person will not be an issue in this thesis, while this situation has always been a challenge for most applications of a vision-based mobile robot.

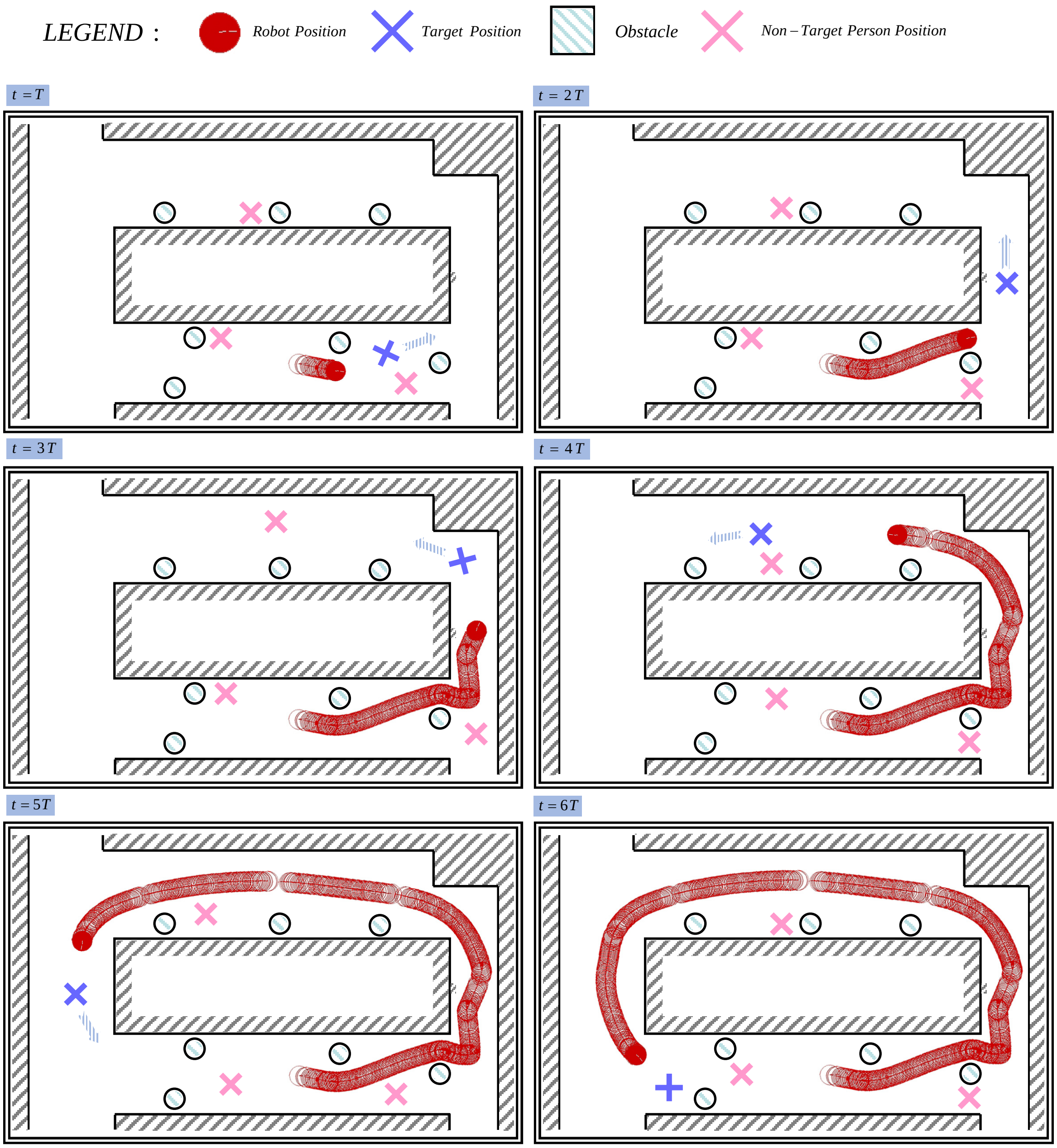

Figure 23. Robot Trajectory in an unstructured environment.

E. SUMMARY

In this chapter, four, main experiments to examine the person-tracking ability are presented. The first experiment is to verify the normal function of the mobile robot using a direct, person-tracking condition without any obstacle between the target person and the robot. The second experiment is to add an obstacle in the person-tracking task and examine the ability of the robot to implement obstacle avoidance and person-tracking simultaneously. The third experiment is to examine the ability of the robot in a common situation, when the robot needs to maintain tracking the target during a turn at a corner. The fourth experiment is based on the examination of the robot's behavior in a general environment, which is unstructured. According to the results of those experiments, the ability of the robot in implementing most general situations is ensured.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION AND FUTURE WORK

A. CONCLUSION

The main objective of this research was to investigate the feasibility of developing a person-tracking robot system using an RF/ultrasonic positioning system. In order to accomplish this objective, the following goals have been achieved in this thesis.

1. Created the interface between the ultrasonic positioning system and the robot system in the operating program.
2. Developed the design of the algorithm that is able to simultaneously avoid obstacles and track the designated person in an unstructured environment.
3. Completed person-tracking experiment when there is no obstacle between the robot and the target person.
4. Completed person-tracking experiment when there is an obstacle between the robot and the target person.
5. Demonstrated person-tracking when the target person makes a turn at a corner.
6. Exhibited person-tracking in an unstructured environment.

During the implementation of the first goal, by using TCP/IP approach, the data developed by the ultrasonic positioning system can be utilized by the robot and analyzed from the remote workstation through the network. The interference issue between the ultrasonic positioning system and the sonar sensor system has been efficiently resolved by sequencing the execution order of those two systems.

The second goal was achieved by designing the potential field algorithm along with the obstacle avoidance algorithm, which was developed from several main situations. In addition, the parameters used in the algorithm have been adjusted during various experiments.

The rest of the goals were to investigate the ability of the mobile robot to accomplish the person-tracking task, utilizing the algorithm designed in Chapter III, which includes person-following and obstacle avoidance. According to the results of several experiments in Chapter IV, the goals have been reached. Since all the goals have been achieved, the feasibility of the main idea in this thesis was verified.

B. FUTURE WORK

Since the feasibility of developing a person-tracking mobile robot using an ultrasonic positioning system in unstructured environments has been ensured, the next step is to improve the efficiency of this system.

The ultrasonic positioning system used in the implementation of a person-tracking mobile robot is a fixed system mounted on the robot. The effective target bearing for the system to detect the target signal is from -90° to 90°. The robot may fail to track the target if the target person intentionally moves out of this range. In addition, while the robot is implementing obstacle avoidance, the increase in the bearing due to the turning of the robot could be a problem. Note that the robot will wander around in the environment until it detects the target signal again.

An active ultrasonic positioning system may be a solution to this problem. The ultrasonic positioning system can maintain the line of sight to the target person and keep tracking the signal even when the robot is completing a turn. Another method to resolve this issue is to include an estimation model of the target person's motion in the algorithm. The Kalman filter is one of the applications. Especially when the target person goes out of the ultrasonic sensor cone, this mechanism provides more accurate information for the robot, so that the robot is able to detect the target signal again in a relatively shorter time interval. Therefore, the overall function of a person-tracking mobile robot can be improved to operate in a more flexible manner.

The ultrasonic positioning system and the sonar sensor system both operate using acoustic principles. Therefore, the robustness of the tracking system in an environment with sound-level noises can be examined in the future work.

APPENDIX

In this appendix, the C++/C code used to operate the overall system is presented as shown below.

```
/***********************************************************
 *                                                         *
 *    PROGRAM: tracking_robot.c                            *
 *                                                         *
 *    PURPOSE: For the robot to follow the specific person and    *
 *                   to implement obstacle avoidance concurrently.   *
 *                                                         *
 *                       Edited by Chuan-Hao Yang          *
 *                                                         *
 ***********************************************************/

/*** Include Files ***/
#include "Nclient.h"
#include <iostream.h>
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netinet/in.h>
#include <math.h>

/* macros to convert Nomad 200 motion commands to Scout motion commands */
#define ROT_CONST 0.118597
#define RIGHT(trans, steer) (trans + (int)((float)steer*ROT_CONST))
#define LEFT(trans, steer)  (trans - (int)((float)steer*ROT_CONST))
#define scout_vm(trans, steer) vm(RIGHT(trans, steer), LEFT(trans, steer), 0)
#define scout_pr(trans, steer) pr(RIGHT(trans, steer), LEFT(trans, steer), 0)

/*** Constants ***/
#define TRUE 1
#define FALSE 0
#define BUFFSIZE 8

/*** Function Prototypes ***/
void GetSensorData(void);
void Movement(void);
void GetUltrasonicData(void);
void Exit(char *mess){perror(mess);exit(0);}
```

```
/*** Global variables ***/
long SonarRange[16];  /* array of sonar readings (inches) */
int BumperHit = 0;    /* boolean value */
/* variable for ultrasonic device connection */
int sock;
struct sockaddr_in ultrasonic_server;
char buffer[BUFFSIZE];
char *ultrasonic_server_ip="192.168.1.15";
char *ultrasonic_server_port="4000";
int distance,angle;


/*** Main Program ***/
main (unsigned int argc, char** argv)
{
  int i, index;
  int oldx, oldy;
  int order[16];

  /* Change the following port number to your own number*/
  SERV_TCP_PORT = 7770;
  /* Connect to Nserver. The parameter passed must always be 1. */
  connect_robot(1);
  /* Initialize Smask and send to robot. Smask is a large array that controls which data the
     robot returns back to the server. This function tells the robot to give us everything. */
  init_mask();

  /* Configure timeout (given in seconds). This is how long the robot will keep moving if
     it becomes disconnected.
  conf_tm(1);

  /* Sonar setup: configure the order in which individual sonar unit fires. In this case, fire
     all units in counter-clockwise order (units are numbered counter-clockwise starting
     with the front sonar as zero). The conf_sn() function takes an integer and an array of
     at most, 16 integers. If less than 16 units are to be used, the list must be terminated by
     an element of value -1. The single integer value passed controls the time delay
     between units in multiples of two milliseconds. Only use the front 5 sonar units in this
     case. */
  for (i=0; i<=2; i++)
     { order[i] = i; }
  for (i=14; i<=15; i++)
     { order[i-11] = i; }
  order[6]= -1;
  conf_sn(25,order);
```

```
  /* Zero the robot. This aligns the turret and steering angles. The repositioning is
       necessary to allow the user to position the robot where it was. */
  oldx = State[34];                    /* remember position */
  oldy = State[35];
  zr();                                /* tell robot to zero itself */
  ws(1,1,1,20);                        /* wait until done with zeroing */
  place_robot(oldx, oldy, 0, 0);  /* reposition simulated robot */


  /* Create connection with Ultrasonic Device */
  if((sock = socket(PF_INET,SOCK_STREAM,IPPROTO_TCP))<0)
  {Exit("Failed to create socket");}
  memset(&ultrasonic_server,0,sizeof(ultrasonic_server));
  ultrasonic_server.sin_family = AF_INET;
  ultrasonic_server.sin_addr.s_addr = inet_addr(ultrasonic_server_ip);
  ultrasonic_server.sin_port = htons(atoi(&*ultrasonic_server_port));
  if(connect(sock,(struct sockaddr *)&ultrasonic_server,sizeof(ultrasonic_server))<0)
  {Exit("Failed to connect with ultrasonic server");}

  /* Main loop. */
  while (!BumperHit)
    {
      GetSensorData();
      GetUltrasonicData();
      Movement();
    }

  /* Disconnect. */
  close(sock);
  disconnect_robot(1);
}


/* Movement(). This function is responsible for using the sensor data to direct the robot's
    motion appropriately. */
void Movement (void)
{
  int m,i;
  int nearsomething_right,nearsomething_left,nearsomething_front;
  int tvel, svel;
  double F_target[2],F_sonar[2],F_total[2];
  int k1 = 15;
  int k2 = 20;
  int k3 = 1;
  int k4 = 10;
  float k5 = 0.75;
```

```
double gain_tvel = 0.1;
double gain_svel = 0.1;
int gain_svel_near_target = 2;
float theta;

/* Make sure we are not about to run into something; check the front sonar sensors. If it
   looks bad, set nearsomething_*.  */
nearsomething_right = FALSE;
nearsomething_left = FALSE;
for (i = 14; i <= 15; i++)
  if (SonarRange[i] < 12 ) nearsomething_right = TRUE;
for (i = 1; i <= 2; i++)
  if (SonarRange[i] < 12 ) nearsomething_left = TRUE;
for (i = 0;i <= 0; i++)
  if (SonarRange[i] < 12 ) nearsomething_front = TRUE;

/* Set limit for the angle */
if (angle > 90) {angle = 90;}
if (angle < -90) {angle = -90;}

/* Compute the attractive force (Equation 3.1~3.8)*/
theta=(360/16)*(3.14/180);
F_target[0] = (distance*cos(angle*3.14/180));
F_target[1] = (distance*sin(angle*3.14/180));
F_sonar[0] = 0;
F_sonar[1] = 0;
for(m=0;m<=2;m++)
{ F_sonar[1] = F_sonar[1]+(SonarRange[m]*sin(m*theta)); }
for(m=14;m<=15;m++)
{ F_sonar[1] = F_sonar[1]+(SonarRange[m]*sin(m*theta)); }
/* Limit for the sonar attractive force */
if (abs((int) F_sonar[1]) >= k5*abs((int) F_target[1]))
{
  F_sonar[1] =(double) k5*abs((int) F_target[1])*(F_sonar[1]/abs((int) F_sonar[1]));
}

F_total[0] = k1*F_target[0]+k3*F_sonar[0];   /* The resultant force in x-axis */
F_total[1] = k2*F_target[1]+k4*F_sonar[1];   /* The resultant force in y-axis */


/* Decide how to move. There are five situations: 1) near target, 2) near something on
   the right, 3) near something on the left, 4) in front of something, 5) clear to move. */
if (distance < 65)  /* Equation 3.11, 3.12 */
  {
    tvel = 0;
    svel = (int) (gain_svel_near_target*F_target[1]);
```

```
    if (abs(angle) <= 10) { svel = 0; }
   }
  else if (nearsomething_right&!nearsomething_left)   /* Equation 3.13 */
   {
    tvel = 0;      /* stop moving, and make a turn. */
    svel = 110;
   }
  else if (nearsomething_left&!nearsomething_right)   /*Equation 3.14 */
   {
    tvel = 0;      /* stop moving, and make a turn. */
    svel = -110;
   }
  else if (nearsomething_front&nearsomething_right&nearsomething_left)
   {                                                   /* Equation 3.15 */
    if (angle >= 0)
     {
      tvel = 0;   /* stop moving, and make a turn. */
      svel = 150;
     }
    else
     {
      tvel = 0;   /* stop moving, and make a turn. */
      svel = -150;
     }
   }
  else  /* it is clear to move */
   {
    svel = (int) (gain_svel*F_total[1]);  /* Equation 3.9 */
    tvel = (int) (gain_tvel*F_total[0]);   /* Equation 3.10 */
   }
  /* limit the translation velocity */
  if(abs(tvel)>120) {tvel=120*tvel/abs(tvel);}

  /* Set the robot's velocities. The first parameter is the robot's translation velocity, in
     tenths of an inch per second. This velocity can be between -240 and 240. The second
     parameter is the steering velocity, in tenths of a degree per second, and can be
     between -450 and 450. */
  scout_vm(tvel,svel);
  printf("%d,%d\n",distance,angle);
}


/* GetSensorData(). Read in sensor data and load into arrays. */
void GetSensorData (void)
{
  int i;
```

```
 /* Read all sensors and load data into State array. */
 gs();

 /* Read State array data and put readings into individual arrays. */
 for (i=0; i<16; i++)
   {
     /* Sonar ranges are given in inches, and can be between 6 and
     255, inclusive. */
     SonarRange[i] = State[17+i];
   }

 /* Check for bumper hit. If a bumper is activated, the corresponding bit in State[33] will
     be turned on. Since we don't care which bumper is hit, we only need to check if
     State[33] is greater than zero. */
 if (State[33]>0)
   {
     BumperHit = 1;
     tk("Ouch.");
     printf("Bumper hit!\n");
   }
}


/* GetUltrasonicData(). Read in ultrasonic data. */
void GetUltrasonicData(void)
{
 int i,j;
 int byte,bytes,n;
 char buff[5*BUFFSIZE];
 while(bytes = recv(sock,buffer,BUFFSIZE-1,0))
 {
   buffer[bytes] = '\0';
   strcpy(buff,buffer);
   for (n = 1;n <= 4;n++)
   {
     byte = recv(sock,buffer,BUFFSIZE-1,0);
     buffer[byte] = '\0';
     bytes = bytes+byte;
     strcat(buff,buffer);
   }
   for (i = 0;i < bytes;i++)
   {
     if (buff[i] == '~')
     {
       for (j = i+1;j < bytes;j++)
       {
```

```
            if (buff[j] == ' ')
            {
                distance = atoi(&buff[i+1]);
                angle = atoi(&buff[j+1]);
                return;
             }
          }
          break;
      }
    }
  }

}
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Xiaoping Yun, and Ko-Cheng Tan. "A Wall-Following Method for Escaping Local Minima in Potential Field Based Motion Planning." *IEEE International Conference on Robotics and Automation*, pp. 421-426, Monterey, CA, 1997.

2. Hung-Kwan Chan. "*The Implementation of a Person Tracking Mobile Robot*." Master's Thesis, The Chinese University of Hong Kong, Hong Kong, China, July 2004.

3. Kai Nickel, Edgar Seemann, and Rainer Stiefelhagen. "3D-Tracking of Head and Hands for Pointing Gesture Recognition in a Human-Robot Interaction Scenario." *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 1-6, 2004.

4. Philipp Althaus, Hiroshi Ishiguro, Takayuki Kanda, Takahiro Miyashita, and Henrik I. Christensen. "Navigation for Human-Robot Interaction Tasks." *Proceeding of the 2004 IEEE International Conference on Robotics & Automation*, pp. 1894-1900, New Orleans, LA, April 2004.

5. B. Kwolek. "Person Following and Mobile Camera Localization Using Particle Filters." *Fourth International Workshop on Robot Motion and Control*, pp. 265-270, June 17-20, 2004.

6. Giorgio Chivilò, Flavio Mezzaro, Antonio Sgorbissa, and Renato Zaccaria. "The Robotic Attendant: How to Follow a Leader Through Visual Servoing." *Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication*, pp. 479-484, Kurashiki, Okayama Japan, September 20-22, 2004.

7. Kanji Tanaka, and Eiji Kondo. "Vision-based Multi-Person Tracking by Using MCMC-PF and RRF in Office Environments." *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 637-642, Sendal, Japan, September 28 - October 2, 2004.

8. Nobutaka Hirai, and Hiroshi Mizoguchi. "Visual Tracking of Human Back and Shoulder for Person-Following Robot." *Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 527-532, 2003.

9. Bogdan Kwolek. "Visual System for Tracking and Interpreting Selected Human Actions." *Journal of WSCG*, Vol. 11, No. 1, ISSN 1213-6972, Plzen, Czech Republic, February 3-7, 2003.

10. Bogdan Kwolek. “Color Vision Based Person Following with a Mobile Robot.” *The International Workshop on Robot Motion and Control*, pp. 375-380, November 9-11, 2002.

11. Hirotaka Ohta, Munemoto Hirako, Kazuhiko Yamamoto, Kunihito Kato, and Kentaro Hayashi. “Development of Entertainment Robot System by Using a Person Detection Method.” *Proceedings of the Seventh International Conference on Virtual Systems and Multimedia*, 2001.

12. Ching-Heng Ku, and Wen-Hsiang Tsai. “Obstacle Avoidance in Person Following for Vision-Based Autonomous Land Vehicle Guidance Using Vehicle Location Estimation and Quadratic Pattern Classifier.” *IEEE Transactions on Industrial Electronics,* Vol. 48, No. 1, February 2001.

13. H. Sidenbladh, D. Kragić, and H. I. Christensen. “A Person Following Behaviour for a Mobile Robot.” *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, pp. 670-675, Detroit, Michigan, May 1999.

14. Tsuyoshi Yamane, Yoshiaki Shirai, and Jun Miura. “Person Tracking by Integrating Optical Flow and Uniform Brightness Regions.” *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, pp. 3267-3272, Leuven, Belgium, May 1998.

15. Maurizio Piaggio, Roberto Fornaro, Alberto Piombo, Luca Sanna, and Renato Zaccaria. “An Optical- Flow Person Following Behaviour.” *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, pp. 301-306, Gaithersburg, MD, September 14-17, 1998.

16. Akihisa Ohya, Yousuke Nagumo, and Youhei Gibo. “Intelligent Escort Robot Moving together with Human - Methods for Human Position Recognition -.” University of Tsukuba, 2002.

17. Kazuyuki Morioka, Joo-Ho Lee, and Hideki Hashimoto. “Physical Agent for Human Following in Intelligent Sensor Network.” *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1234-1239, EPFL, Lausanne, Switzerland, October 2002.

18. Kazuyuki Morioka, Joo-Ho Lee, and Hideki Hashimoto. “Human-Following Mobile Robot in a Distributed Intelligent Sensor Network.” *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 1, February 2004.

19. Sebastian Lang, Marcus Kleinehagenbrock, Sascha Hohenner, Jannik Fritsch, Gernot A. Fink, and Gerhard Sagerer. “Providing the Basis for Human-Robot-Interaction: A Multi-Model Attention System for a Mobile Robot.” *ICMI’03*, pp. 28-35, Vancouver, British Columbia, Canada, November 5-7, 2003.

20. Nomadic Technologies, Inc. “Nomad Scout Language Reference Manual.” Technical report, 2133 Leghorn Street, Mountain View, CA 94043, July 12, 1999.

21. Nomadic Technologies, Inc. “Nomad Scout User’s Manual.” Technical report, 2133 Leghorn Street, Mountain View, CA 94043, July 12, 1999.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Code EC
   Department of Electrical and Computing Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor Xiaoping Yun
   Department of Electrical and Computing Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Marcello Romano
   Department of Mechanical and Astronautical Engineering
   Naval Postgraduate School
   Monterey, California

6. Chuan-Hao Yang
   YongKang, Tainan710
   Taiwan